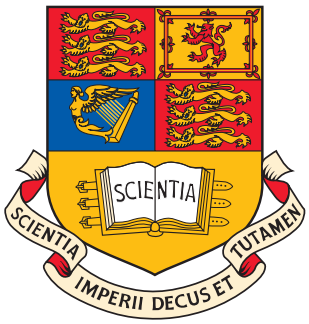
Imperial College London
Department of Department of Earth Science and Engineering

Discontinuous Galerkin Methods for Computational Radiation Transport

Simon Richard Merton

June 2012



Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computational Physics
of Imperial College London
and the Diploma of Imperial College London

I begin this PhD knowing very little about a great deal...

Preface

Declaration of Originality

The material in this Thesis is entirely the result of my own independent research under the supervision of Professor Christopher Pain and Professor Richard Smedley-Stevenson. I declare the research is original except where indicated by reference and that all published or unpublished material used in this Thesis has been given full acknowledgement. Any views expressed in this Thesis are those of the author and in no way represent those of Imperial College.

Simon R. Merton

(October 2011)

Author Contact Details

Email: simon.merton@imperial.ac.uk

Research URL: http://amcg.ese.ic.ac.uk/index.php?title=Simon_Merton

Postal Address: Department of Earth Science and Engineering,
Imperial College London,
South Kensington Campus,
Exhibition Road,
London, SW7 2AZ, UK.

I dedicate this to my mother, father and brother, I did it for you.

Abstract

This Thesis demonstrates advanced new discretisation technologies that improve the accuracy and stability of the discontinuous Galerkin finite element method applied to the Boltzmann Transport Equation, describing the advective transport of neutral particles such as photons and neutrons within a domain. The discontinuous Galerkin method in its standard form is susceptible to oscillation detrimental to the solution. The discretisation schemes presented in this Thesis enhance the basic form with linear and non-linear Petrov Galerkin methods that remove these oscillations. The new schemes are complemented by an adjoint-based error recovery technique that improves the standard solution when applied to goal-based functional and eigenvalue problems. The chapters in this Thesis have been structured to be submitted individually for journal publication, and are arranged as follows. Chapter 1 outlines the Thesis and contains a brief literature review. Chapter 2 introduces the underlying space-angle discretisation method used in the work, and discusses a series of potential modifications to the standard discontinuous Galerkin method. These differ in how the upwinding is performed on the element boundary, and comprise an upwind-average method, a Petrov-Galerkin method that removes oscillation by adding artificial diffusion internal to an element and a more sophisticated Petrov-Galerkin scheme that adds dissipation in the coupling between each element. These schemes are tested in one-dimension and Taylor analysis of their convergence rate is included. The chapter concludes with selection of one of the schemes to be developed in the next part of the Thesis. Chapter 3 develops the selected method extending it to multi-dimensions. The result is a new discontinuous Petrov-Galerkin method that is residual based and removes unwanted oscillation from the transport solution by adding numerical dissipation internal to an element. The method uses a common length scale in the upwind term for all elements. This is not always satisfactory, however, as it gives the same magnitude and type of dissipation everywhere in the domain. The chapter concludes by recommending some form of non-linearity be included to address this issue. Chapter

4 adds non-linearity to the scheme. This projects the streamline direction, in which the dissipation acts, onto the solution gradient direction. It defines locally the optimal amount of dissipation needed in the discretisation. The non-linear scheme is tested on a variety of steady-state and time-dependent transport problems. Chapters 5, 6 and 7 develop an adjoint-based error measure to complement the scheme in functional and eigenvalue problems. This is done by deriving an approximation to the error in the the bulk functional or eigenvalue, and then removing it from the calculated value in a post-process defect iteration. This is shown to dramatically accelerate mesh convergence of the goal-based functional or eigenvalue. Chapter 8 concludes the Thesis with recommendations for a further plan of work.

Publications

Peer Reviewed Publications

- Optimal Discontinuous Finite Element Methods for the Boltzmann Transport Equation with Arbitrary Discretisation in Angle (Merton et al., 2008)

- The Inner Element Sub-Grid Scale Finite Element Method for the Boltzmann Transport Equation (Buchan et al., 2010)

- Riemann Boundary Conditions for the Boltzmann Transport Equation using Arbitrary Angular Approximations (Buchan et al., 2011)

Conference Proceedings

- A Non-Linear Optimal Discontinuous Petrov-Galerkin Method for Stabilising the Solution of the Transport Equation (Merton, 2009)

- Adjoint Based Error Measures for Functional Defect Correction in Deterministic Neutron Transport Applications (Merton, 2010)

- A Non-Linear Optimal Discontinuous Petrov-Galerkin Method for Stabilising the Solution of the Transport Equation (Merton, 2011)

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Synopsis

A PhD Thesis at Imperial College, London is proposed concerned with the discontinuous solution of the Boltzmann Transport Equation (BTE). The project is to be in collaboration between Imperial College and AWE plc and is well aligned with the current research interests of both the Earth Science and Engineering department at Imperial College and the Computational Physics Group at AWE. The focus of the Thesis is the discontinuous Galerkin (DG) method and the development of new spatial discretisation schemes within the DG framework for the solution of the transport equation. This is to be supplemented with methods for controlling errors in functionals and eigenvalues in transport problems. These methods are arbitrary angle; that is they are implemented within a finite element Riemann framework that supports any angular discretisation scheme.

Contents

1.1. Scope

The focus of this Thesis, undertaken at Imperial College London, is on the discontinuous Galerkin finite element method applied to the Boltzmann Transport Equation (BTE) in the context of computational neutron and radiation transport. The research represents an ongoing collaboration between Imperial College London and AWE plc that has been built up over many years. The development of the finite element method for computational neutron transport and radiation modelling is central to research programmes within both the Applied Modelling and Computation (AMCG) group at Imperial College and the Computational Physics Group at AWE. The research topic of this Thesis is the discontinuous Galerkin (DG) finite element method and how this can be developed further. The Thesis develops new spatial discretisation schemes within the DG framework for the solution of the transport equation. These contribute to the development of Petrov-Galerkin methods for transport applications. The new schemes are supplemented with methods for controlling errors in the bulk functionals and eigenvalues that arise in transport problems, such as reactor criticality calculations and detector response calculations. These methods use the adjoint system of equations to derive approximations to the error in the functional or eigenvalue so these quantities can be improved. This is ultimately motivated by improving the numerical efficiency of large scale problems. The methods that are developed in this Thesis are arbitrary angle; that is they are implemented within a very general finite element Riemann framework that supports any angular discretisation scheme.

1.2. Overview

The first part of this work is to further the development of DG as a discretisation technology, by adding an enhancement to the underlying standard form for the removal of unwanted oscillatory behaviour in the transport solution. The standard

DG method is a popular choice for transport problems, however it is prone to oscillation and some form of error control is needed. In the second part of the Thesis, methods for improving the mesh convergence of functionals and eigenvalues are considered. This is done by an enrichment of the solution. The Thesis is arranged as follows. The present chapter contains some background on the present work. This includes a literature review in acknowledgement of the accomplishments in this field, and related topics. Following this, the notation used in the present work is introduced in chapter (2). This includes details of the spatial and angular discretisation schemes. An account of some one-dimensional DG schemes is given, differing in how the elements are coupled across the mesh. The one-dimensional schemes include an averaging of the information on the boundary of the elements, the introduction of numerical diffusion into the discretisation for stability and the addition of a residual-based term into the discretisation for numerical stability. The standard DG method is also included for benchmarking. Taylor analysis is used to quantify the theoretical convergence rate that each scheme has, and compared with the actual convergence rate of each method in practise. The chapter concludes with selection of one of the schemes as the topic of further work. This includes extension of the scheme to multi-dimensions and further development. Chapter (3) develops the chosen scheme formulating a new discretisation method that adds an optimal amount of dissipation for numerical stability. This is shown to improve the behaviour of the solution with respect to the standard DG method. This approach is linear however, since the same element length scale is used resulting in the same magnitude and type of numerical dissipation throughout the solution domain.
Chapter (4) improves on this by adding non-linearity to the upwind term so that the amount and type of dissipation is adjusted locally. Results are demonstrated for the discrete ordinates (SN), spherical harmonics (PN) and wavelet (LWN) angular framework. Chapter (5) develops a scheme for improving functional estimates. This is done by a post-processing of the DG solution to improve smoothness locally. An approximation to the error in the functional is then derived by convolving the adjoint equations with the residual. This approximation to the error is then removed from the functional estimate. This is shown to improve convergence rate of the functional considerably, and performs consistently well in different angular discretisation frameworks. Chapter (6) develops this adjoint-based technique so it may be used to recover the error in the K-eigenvalue in elliptic problems. Chapter (7) demonstrates adjoint-based error recovery schemes for eigenvalues in hyperbolic transport problems, discussing various schemes that smooth the trans-

port solution after it has been obtained from an inverse power iteration. In chapter (8) conclusions are drawn with recommendations for further work.

1.3. Literature Survey

An overview of the history of approximate methods is given by Zienkiewicz (Zienkiewicz, 2005), where it is illustrated how the works of Rayleigh and Ritz on variational methods combine in the early 20^{th} century with the works of Gauss, Galerkin and Biezeno on weighted residuals to form piecewise continuous trial functions. These are regarded as the foundation of the finite element method as it is recognised in the present day.

The advent of high performance computing resulted in finite element schemes being applied to criticality modelling and radiation transport (Hill, 1975; Seed et al., 1977). These new finite element methods offered flexibility that finite difference schemes lacked. They provided much greater freedom in where to locate the spatial nodes defining the shape of an element. This allowed the first accurate modelling of curved material interfaces (Reed and Hill, 1973) and also of hexagonal geometries (Miller et al., 1973). Ohnishi and co-workers were among the first to demonstrate how actual reactor geometries could be represented more accurately with triangular finite element meshes than they could be with finite difference meshes composed of simple rectangular cells (Ohnishi, 1971). Further contributions included those of Miller and co-workers, who successfully combined a finite element that was bilinear in angle with linear and bilinear spatial elements (Miller et al., 1973). They applied this novel element to a reactor-like problem that was difficult to calculate with conventional finite difference schemes due to the orientation of material boundaries. Accurate modelling of the interfaces required a finite difference mesh that was computationally prohibitive. Using triangular finite elements, a much coarser spatial mesh could accurately represent the interfaces. Novel element types such as these led to the development of unstructured meshes for accurately represented boundaries (Eaton et al., 2003). These meshes are not possible using diamond difference methods, but are of great importance in modern transport calculations of three dimensional systems (Eaton, 2004). Wareing has demonstrated the accuracy that unstructured tetrahedral and hexahedral meshes can achieve in three-dimensional radiation transport applications (Wareing et al., 2001).

Another notable deficiency of diamond difference methods is their tendency to

oscillate producing negative fluxes. This is of concern in any radiation transport scheme since it can make the solution unstable. This is particularly apparent in coupled calculations, for example where the fluxes are driving an in-line depletion calculation (burn-up) or being used to calculate energy subsequently deposited into the system. It has been shown that finite element schemes are in general less likely to produce negative fluxes than central difference diamond schemes, due to their lower spatial truncation error (Alcouffe, 2003). However, the use of finite elements does not eliminate occurrence of negative fluxes and hence there is a need for some form of stabilisation. Traditionally this is achieved in diamond difference codes by means of negative flux fix-up (Lewis and Miller, 1993). This is a large computational overhead, and degrades accuracy of the solver to a degree that may not be acceptable. This has motivated the development of more sophisticated weighted diamond schemes (Morel, 1984), which could be parametrically tuned to particular cross-section regimes. Once tuned, such schemes guarantee positive solutions. By combining several such schemes that were complementary to one another, Reed was able to produce a differencing method totally free from spatial oscillation (Reed, 1971).

In addition to spatial oscillation, instabilities can occur in angle. This is most pronounced in the S_N (discrete ordinates) scheme where characteristic rays imprint on the solution. It has been realised (Morel, 2003) that ray effects arise from the lack of rotational invariance of the streaming operator in the transport equation. One can therefore avoid them by using a rotationally invariant angular discretisation, or by making the equations invariant to rotation by placing the off-diagonal elements of the streaming matrix in the source term and handling this iteratively (Sanchez, 2011). Many early finite element transport codes used the S_N method to difference in angle, for example ONETRAN (Hill, 1975), and early two-dimensional codes such as TWOTRAN which suffered badly from ray-effects. Much work has been done on various ray-effect palliatives to mitigate this problem (Lathrop, 1971; Morel, 2003), and on alternative angular differencing schemes. Ohnishi and co-workers demonstrated how applying finite element techniques to model the angular variation, as an alternative to discrete ordinates, could totally eliminate the ray-effects seen on all discrete ordinates results (Ohnishi, 1971). Although their scheme was continuous in space, perhaps because methods to handle spatial derivatives across the discontinuities were in their infancy, making it discontinuous in angle was achievable since the Cartesian (x-y) form of the transport equation was used in which no angular derivatives appear. Martin and Duderstadt also made

use of a finite element that was discontinuous in angle (Martin and Duderstadt, 1977). Realising this would decouple their solution into forward and backward components, they were able to eliminate unphysical fluxes in voids. Their endeavour overlapped with the work of Briggs et al (Briggs and Miller, 1975), who were among the first to discover that the finite element approximation in angle resembled an elliptic discrete ordinate-like problem, the discrete ordinate scheme itself being hyperbolic. This is exemplified by angular smearing of the streaming operator, and Briggs was able to show that at least for second order forms of the transport equation, it would remove all ray characteristics along the directions of neutron travel. Martin and Duderstadt were able to show this could also be the case for first order forms of the equation (Martin and Duderstadt, 1977).

The use of spherical harmonics (P_N) to expand the angular flux provides another alternative angular differencing scheme (Caldiera, 2005). This was first applied computationally in one-dimensional problems when difficulties were encountered with the S_N method (Davison, 1957). It was later adapted for multi-dimensional systems by Fletcher and co-workers (Fletcher, 1983). Although totally free from ray-like oscillation, the P_N method being a finite approximation also suffers from serious defects. These manifest as dispersive undershoot and dissipative overshoot (Gibbs oscillation) and can be highly problematic for P_N codes in time-dependent transport (Yildiz, 2000). Like the ray-effect that occurs with the S_N approximation, these oscillations reduce as the order of the expansion is increased. However, different regions of a problem tend to require different angular resolutions. Such requirements can vary with time also, and it is therefore useful to be able to adapt in angle. This presents a challenge with the deterministic methods since they are not hierarchical. For example, an S_N expansion can not easily be extended without modification to the existing terms. Walsh functions may be used to transform the S_N equations so that they become hierarchical. Buchan has demonstrated how this can be used to form adaptive angular schemes (Buchan, 2003a). Buchan has since developed linear and quadratic spherical wavelet methods to demonstrate this on the sphere (Buchan, 2003b,c), and more recently linear and quadratic octahedral wavelets (Buchan, 2005a). The quadrature points in these wavelet schemes are defined on the surface of hierarchical shapes rather than on the unit sphere, and so lend themselves well to adaptivity.

As finite element schemes became more sophisticated they were applied to increasingly demanding problems in radiation transport. It became a requirement for a scheme to track and accurately resolve abruptly changing solution fields

while remaining free from oscillation. This required two conflicting properties for a scheme to have; clearly, there needed to be some rapid response to changes in the characteristics of the solution, but without the scheme producing unphysical behaviour on recovery and without the response destabilising the differencing scheme. Methods for achieving this have been developed by Pain and co-workers (Pain and Goddard, 2000) where a scheme using a dual set of basis functions was produced, along with a characteristic discontinuous finite element method (DFEM) that used an optimal amount of upwinding for stability. These schemes were able to accurately represent solution fields that were almost discontinuous in space. The latter applied a positive diffusion with optimal weighting of the upwind term to match analytical solutions, and has motivated the work in the present paper.

Discontinuity capturing schemes that were originally developed for the modelling of incompressible flow in fluid mechanics (Brooks and Hughes, 1982) have been adapted for use in radiation transport. Related schemes include the discontinuous Galerkin (DG) scheme, in which continuity of the solution field is not enforced allowing sharp gradients in the radiation flux to be captured. This becomes important where sudden changes in cross-section occur, for example at material interfaces, and is thus necessary for a scheme to function across changing radiation regimes. The requirement to capture discontinuities without destabilising the differencing scheme has resulted in the development of Petrov-Galerkin (PG) methods for radiation transport. Pain and Eaton have developed streamline upwind Petrov-Galerkin (SUPG) methods (Pain et al., 2006a), which differ from conventional shock-capture schemes in that they introduce the artificial dissipation in the direction of the streamline. Following the successful deployment of these schemes for the steady-state Boltzmann transport equation (Pain et al., 2006a), they were subsequently applied in space-time (Pain et al., 2006b). The boundary conditions in these schemes are commonly enforced using a Riemann approach. In widespread use in fluid mechanics (Toro, 1997), Riemann discretisation methods were adapted by Eaton for use in advective radiation transport (Eaton et al., 2003; Eaton, 2004), who demonstrated the ease with which they can be applied to multi-dimensional unstructured meshes, and also by Holloway and co workers (Holloway and Brunner, 2005; McClarren and Holloway, 2007). In addition, Holloway and McClarren have shown that asymptotic diffusion limits may be established for these new Riemann methods (McClarren and Holloway, 2005).

The Discontinuous Galerkin (DG) method was first applied computationally in 1973 by Reed and Hill (Reed and Hill, 1973) at Los Alamos, as a means of spa-

tially discretising the neutron transport equation when difficulties were encountered with traditional (continuous) representations. First use of the scheme was on triangular meshes to solve reactor-like geometries difficult to calculate on conventional meshes, and to improve representation of curved material interfaces. In 1974, LeSaint and Raviart (LeSaint and Raviart, 1974) performed the first numerical analysis of the DG method and it immediately became popular in various areas of modelling besides neutron transport, notably shock wave propagation in elastic media (Wellford and Oden, 1975), visco-elastic flow (Fortin and Fortin, 1989) and error control (Delfour and Trochu, 1978). It was applied to time dependent parabolic systems by Jamet and co-workers (Jamet, 1978), to hyperbolic equations by Hulbert in 1978 (Hulbert and Hughes, 1990), to the Navier-Stokes equations (Lomtev and Karniadakis, 1999) and to magneto-hydrodynamics by Warburton and Karniadakis (Warburton and Karniadakis, 1999). More recently, the scheme has become popular in mainstream computational fluid dynamics. Lending itself very well to adaptivity, simplicity of implementation and numerical robustness, the DG method is of great importance in many areas of modelling. Continuity of solution is not enforced, allowing the capture of sharp gradients in the solution field. The DG method has been used for the temporal discretisation of the Boltzmann Transport Equation (BTE) in combination with a DG spatial discretisation allowing the capture of radiation fields that are extremely poorly behaved.

Although adequate for a broad range of problems, the DG method can in certain cases produce unwanted oscillation in the transport solution. In radiation transport applications, oscillations occur when the flux gradient is sharp, for example at a boundary of very dissimilar material properties. In fluid applications, shocks occur when a succession of compression waves combine disrupting the flow variables. This has motivated development of methods such as the Petrov-Galerkin (PG) method for improved handling of discontinuities, and smoothing solutions in shocked high speed flows (Hughes and Tezduyar, 1984). This works by adding artificial dissipation internal to an element, typically in the streamline or solution gradient direction (Beau and Tezduyar, 1991). The original upwind approach was first introduced by Christie and co-workers (I. Christie and Zeinkiewicz, 1976). This method worked by modifying the finite element stencil in order to achieve an upwind effect. Essentially this entails weighting the element upstream of a node more so than the downstream element, smoothing solutions with an artificial dissipation or viscosity. Although very successful in one dimension, in multi-dimensions these schemes often result in an excessive amount of dissipation being

introduced to the solution field in directions perpendicular to the flow. Thus they received much criticism from the finite element community. By restricting the dissipation so it acted only in the flow direction, Brooks and Hughes introduced the Streamline[1] Upwind Petrov-Galerkin (SUPG) method in the form that it is recognised in the present day (Brooks and Hughes, 1982). This method brings us all the advantages of the traditional upwind methods, but without the difficulties of criticism initially received by the artificial diffusion concept.

Methods such as SUPG have recently been applied for gradient control in transport calculations (Pain et al., 2006b,a). However, methods that add diffusion only in the streamline direction generally cannot remove shocks, as these require the dissipation to act in the direction of the solution gradient. This was first realised by Hughes and co-workers, who proposed adding an additional discontinuity capturing term on to the existing SUPG formulation in order to rectify the remaining shortcomings (Hughes et al., 1986). This was generalised to multi-dimensional advection diffusion problems also by Hughes (Hughes et al., 1985, 1987), where more basic methods failed due to excess dissipation being introduced even in one dimension. More recently, further work has been done on implementing shock capture operators within modern SUPG formulations (Beau and Tezduyar, 1991). SUPG formulations have been integrated into space-time Riemann solvers, providing an excellent solution of hyperbolic systems such as the Euler Equation and the Boltzmann Transport Equation. Recently, these have been used very successfully where steep gradients dominate the solution field (Eaton, 2004; Holloway and Brunner, 2005; McClarren et al., 2005).

Finite element schemes are computationally more intensive than the simpler diamond schemes, due to the fact they contain more unknowns (and therefore more equations) per mesh cell . However, finite element schemes have mesh convergence rates superior to diamond schemes when applied to spatial discretisations of the transport equation (Adams, 1998; Alcouffe et al., 1979). This has been shown also for bilinear discontinuous discretisations of the time-dependent transport equation (Warsa and Prinja, 1999). Even in axisymmetric geometry, discontinuous finite elements exhibit third order for mesh sizes < 1 mfp (Alcouffe, 2003). This is in agreement with the findings of Wareing et al (Wareing et al., 2001) whose discontinuous method also exhibited third order behaviour, even on non-orthogonal hexahedral meshes with three spatial dimensions. In contrast, diamond methods typically exhibit only second order accuracy, perhaps even less in high in-

[1] in transport problems, the "streamline" refers to the direction in which particle advection occurs.

stances of negative flux fix-ups. It has been suggested that finite element schemes may therefore translate into efficiency savings over diamond schemes. Coarser spatial finite element meshes give a more acceptable error due to their improved truncation order. In addition, robust convergent differencing may be established in the thick diffusion limit for linear discontinuous methods in multi-dimensions, further improving their accuracy (Borgers et al., 1992; Larsen et al., 1987; Larsen and Morel, 1989) and extending their application (Morel et al., 1996). Efficient diffusion synthetic acceleration (DSA) schemes are often a necessary component of finite element solvers, especially for solvers using source iteration in multi-dimensions (Morel et al., 1993). However, these are not considered in this Thesis.

CHAPTER 2

BACKGROUND

Synopsis

This section introduces the Boltzmann Transport Equation in the context of computational radiation transport modelling, an application for which it has become increasing popular since the discontinuous Galerkin method was first suggested in 1973. The space-angle discretisation scheme used in the Thesis is introduced in this chapter and also the finite element Riemann method. This is the method that is used for defining the surface integrals in the equations. The finite element Riemann method is very general and allows the schemes that follow to be demonstrated in different angular expansion frameworks, that use different basis functions in direction of particle travel. Some one-dimensional discontinuous methods are discussed in this chapter, one of which is selected for further development. The method that is selected forms the basis of the chapter that follows.

Contents

2.1. Introduction

Numerical solutions to the time-dependent form of the Boltzmann equation are of great importance in a variety of neutron transport and radiation-hydrodynamics computational problems. Applications range from representing the transient behaviour of reactors and the time-dependent simulation of delayed neutron physics to the modelling of fissile solutions. Temporal discretisation technologies have become as sophisticated as schemes once only considered for spatial domains. The DG method, in particular, is becoming an increasingly popular choice for space-time differencing of the BTE (Cheng and Shu, 2007). This allows capture of radiation fields that are extremely poorly behaved. The DG method may be used on both the spatial and temporal grids, treating time as just another dimension over which the transport takes place. In the case of two spatial dimensions, the resulting element has the same number of unknowns representing the space-time variation as there are in a three-dimensional element of the same polyhedral type and order. This allows the solution to be discontinuous on both the spatial and temporal boundaries. The Boltzmann Transport Equation (BTE) is introduced in section (2.2) and the notation that is used in the present work is established. This is followed by a description in section (2.3) of the angular expansions that are used. Three bases in direction of particle travel are considered, and they comprise the

discrete ordinates method (S_N), the spherical harmonics method (P_N) and linear octahedral wavelets (LW_N). This is followed in section (2.4) by an introduction of the discretisation techniques used on the spatial and temporal grids for the projection of the angular solution to the BTE. The methods developed in the chapters that follow are demonstrated with all three angular schemes. This is to ensure they may be used within different angular expansion frameworks and do not exclude the use of any particular angular scheme. In order for the methods to work with all these angular schemes, boundary conditions have been implemented in a general manner. This is achieved by coupling the elements in space-time using a finite element Riemann approach discussed in section (2.5). Coupling elements in space has traditionally been a challenge for schemes whose equations are not diagonal in normal space. In the case of spherical harmonics (P_N) (Eaton, 2004; Fletcher, 1983; Yildiz, 2000; Holloway and Brunner, 2005), directions in which information travels between elements is not easily defined as it is in the discrete ordinates (S_N) approximation. The difficulty arises because directions are not discrete making the coupling between neighbouring elements non-trivial. To distinguish between incoming and outgoing information, half-range integrals have traditionally been used and these are computationally expensive to calculate, particularly for elements of mixed orientation with respect to the coordinate axes of the solution domain. Conversely, the S_N equations are diagonal in normal space making it straight forward to distinguish outgoing information from incoming, allowing elements to be coupled with very little computational overhead. The Riemann method implemented here involves diagonalising each angular Jacobian matrix [1] and determining the left and right eigenvectors and eigenvalues. These are used to resolve the integrals on the element surfaces. The mesh-angle sweeping method on which the solution algorithms are based is introduced in section (2.6). The method of source-iteration that one might implement in order to solve the multi-group transport equation is introduced in section (2.7).

Some one-dimensional discontinuous spatial differencing schemes are discussed in section (2.8) which differ in how the incoming information is up-winded on the element boundaries. The methods include the standard DG scheme, an up-wind average scheme, a scheme using sub-grid scale bubble functions to enhance the element and two types of Petrov-Galerkin stabilisation schemes. One of the

[1] these are the matrices containing the Cartesian components of the directions in which particles can move, and are referred to elsewhere in the literature (Eaton et al., 2003; Pain et al., 2006b) as the "Jacobians" of the angle of particle travel. This terminology has been adopted throughout this Thesis.

Petrov-Galerkin schemes adds the artificial dissipation internal to the element and the other adds the dissipation both internal to the element and also into the coupling term between the elements. The one-dimensional schemes are tested using Taylor analysis in section (2.9). This analysis suggests their truncation order. The methods are compared in simple numerical problems in section (2.10). Section (2.11) concludes the chapter with one of the schemes being selected as the topic of further work.

2.2. Boltzmann Transport Equation

The advective transport of neutral particles, such as photons or neutrons, through a material or vacuum in multi-dimensional Cartesian space is commonly described by the time-dependent, linear form of the one-speed Boltzmann Transport Equation (BTE). This equation may be written as:

$$\begin{aligned}\left(\frac{1}{v}\frac{\partial}{\partial t} + \mathbf{\Omega}\cdot\nabla + \sigma_t(\mathbf{r},E)\right)\psi(\mathbf{r},\mathbf{\Omega},E) &= q(\mathbf{r},\mathbf{\Omega},E) \\ &= q_{bdy}(\mathbf{r},\mathbf{\Omega},E) \\ &+ q_{fis}(\mathbf{r},E) \\ &+ q_{sca}(\mathbf{r},\mathbf{\Omega},E), \qquad (2.1)\end{aligned}$$

where v refers to the speed of the neutral particles travelling with energy E in the direction $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$, and $\psi(\mathbf{r},\mathbf{\Omega},E)$ is the angular flux at position $\mathbf{r} = (t,x,y,z)^T$ in space-time, to which computational solutions are required. The quantity $\sigma_t(\mathbf{r},E)$ is the total cross-section. This describes losses due to absorption in the host media. It is common practice in radiation transport literature to make the assumption that the material cross-sections are time-independent. In general this will not be the case, due to material properties varying with time, for example if effects such as nuclear depletion are calculated. This requires the material cross-sections to vary in space-time. In the present work, it is assumed that $\sigma_t(\mathbf{r},E)$ is constant with respect to time. $q(\mathbf{r},\mathbf{\Omega},E)$ is the sum of all sources driving the system which may include imposed body sources, fission, out of group scattering and angular scattering sources. These are represented in equation (2.1) as $q_{bdy}(\mathbf{r},\mathbf{\Omega},E)$, $q_{fis}(\mathbf{r},\mathbf{\Omega},E)$ and $q_{sca}(\mathbf{r},\mathbf{\Omega},E)$ respectively. The fission source

is defined as:

$$q_{fis}(\mathbf{r}, E) = \int_{\Omega} d\Omega \nu(E) \sigma_f(\mathbf{r}, E) \chi(E) \psi(\mathbf{r}, \mathbf{\Omega}, E), \tag{2.2}$$

in which $\nu(E)$ is the mean number of neutrons that are produced in a fission that is caused by a neutron with energy E, $\sigma_f(\mathbf{r}, E)$ is the fission cross-section at position $\mathbf{r}$ for neutrons with energy E and $\chi(E)$ is the fission spectrum. The scattering source is defined as:

$$q_{sca}(\mathbf{r}, \mathbf{\Omega}, E) = \int_{\Omega'} d\Omega' \int_{E'} dE' \sigma_s(\mathbf{r}, \mathbf{\Omega}' \rightarrow \mathbf{\Omega}, E' \rightarrow E) \psi(\mathbf{r}, \mathbf{\Omega}', E'), \tag{2.3}$$

where $\sigma_s(\mathbf{r}, \mathbf{\Omega}' \rightarrow \mathbf{\Omega}, E' \rightarrow E)$ is the cross-section of a particle initially travelling in direction $\mathbf{\Omega}'$ with energy E' scattered into direction $\mathbf{\Omega}$ and into energy E by the host media. The angular flux emission due to such scattering events results in this extra source term. In general, these sources vary in time also. In the current work, scattering is assumed to be isotropic. The inclusion of anisotropy on the scattering term is a straightforward extension to the present work. The scattering kernel $\sigma_s(\mathbf{r}, \mathbf{\Omega}' \rightarrow \mathbf{\Omega}, E' \rightarrow E)$ is often written as $\sigma_s(\mathbf{r}, \mathbf{\Omega}' \cdot \mathbf{\Omega}, E' \rightarrow E)$, in which the dot product $\mathbf{\Omega}' \cdot \mathbf{\Omega}$ defines the angle between the trajectory of the incident particle and the scattered particle. The angle by which the particle is scattered depends only on the angle of incidence. This is due to the rotational invariance of the material. The scattering kernel is then expanded in Legendre polynomials, as discussed in section (2.3), to represent the angular distribution independently of the approximation used to represent the angular flux itself. The direction $\mathbf{\Omega}$ is a vector that defines a point on the surface of the unit sphere, and may be expressed in terms of Cartesian components $(\Omega_x, \Omega_y, \Omega_z)^T$. This represents all directions of particle travel inside the unit sphere. In order to solve equation (2.1) on a computational mesh to obtain numerical solutions to $\psi(\mathbf{r}, \mathbf{\Omega}, E)$, the angular flux must be discretised in time, space, angle and in energy. The equation has a very large phase-space, due to the large number of independent variables. These comprise three spatial coordinates, one time coordinate, one energy coordinate and the two components of the angular coordinate; seven in total. It is usual to perform the angular discretisation first, followed by the spatial and temporal differencing for which linear discontinuous finite elements are used in the current work. This results in a non-symmetric system of coupled hyperbolic equations one can solve by

matrix inversion. The current work omits the energy discretisation from consideration; this is left as a future topic.

2.3. Angular Discretisation

It is usual to perform the angular discretisation before any spatial and temporal discretisation is considered. To begin, one must select a suitable set of basis functions in direction of particle travel. In the present section, no assumption is made about these functions, they are to be considered arbitrary. In the present work, the energy dependence of the solution is omitted from consideration and the angular discretised form of the transport equation (2.1) is obtained by expanding the angular variable and generating the weak form equations through a weighted residual method. That is the angular flux $\psi(\mathbf{r}, \mathbf{\Omega})$ is approximated by the expansion:

$$\psi(\mathbf{r}, \mathbf{\Omega}) \approx \sum_{j=1}^{\mathcal{M}} \mathcal{G}_j(\mathbf{\Omega}) \Psi_j(\mathbf{r}), \tag{2.4}$$

where $\mathcal{G}_j(\mathbf{\Omega})$ represent the expansion's spatially invariant angular basis functions and $\Psi_j(r, t)$ are their corresponding angular moments. The approximation (2.4) is inserted into the transport equation (2.1) and a system of $\mathcal{M}$ equations are created by multiplying equation (2.1) with a weighting function $\mathcal{G}'_i(\mathbf{\Omega})$, for $i = 1, \mathcal{M}$, and integrating the resulting equation over the angular variable Ω. In this work the Galerkin method is applied in angle. This method uses the angular basis functions in the expansion (2.4) as the weighting functions, i.e. $\mathcal{G}' = \mathcal{G}$:

$$\int_{4\pi} d\Omega \mathcal{G}_i(\mathbf{\Omega}) \left\{ \left(\frac{1}{v} \frac{\partial}{\partial t} + \mathbf{\Omega} \cdot \nabla + \sigma_t(\mathbf{r}) \right) \sum_{j=1}^{\mathcal{M}} \Psi_j(\mathbf{r}) \mathcal{G}_j(\mathbf{\Omega}) \right.$$
$$\left. - q_{ex}(\mathbf{r}, \mathbf{\Omega}) + q_s(\mathbf{r}, \mathbf{\Omega}) \right\} = 0,$$
$$\text{for } i = 1, 2, \ldots, \mathcal{M}. \tag{2.5}$$

From here on, the vector $\mathbf{r}$ is described by the Cartesian coordinates $(x, y, z)^T$ and the angular vector $\mathbf{\Omega}$ is described by the variables μ , the cosine of the polar angle θ measured with respect to the spatially fixed z axis, and the azimuthal angle ω, the angle describing the direction of Ω in the xy plane. This set of spatial and angle coordinates is illustrated in figure (2.1), and allows the streaming operator

of equation (2.5) to be defined specifically as:

$$\mathbf{\Omega} \cdot \nabla = (\Omega_x, \Omega_y, \Omega_z) \cdot \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right) = \Omega_x \frac{\partial}{\partial x} + \Omega_y \frac{\partial}{\partial y} + \Omega_z \frac{\partial}{\partial z}, \qquad (2.6)$$

where Ω_x, Ω_y and Ω_z are components of the unit vector describing the direction of Ω in Cartesian space. Figure (2.1) shows two Cartesian coordinate systems; one with abscissae (x, y, z) containing the spatial coordinates of a particle, and another with abscissae (e_x, e_y, e_z) containing the angular coordinates $(\Omega_x, \Omega_y, \Omega_z)$ that define the direction $\Omega(\theta, \omega)$ in which the particle is travelling. The particle is illustrated at position P and is denoted by a solid circle. The direction vector along

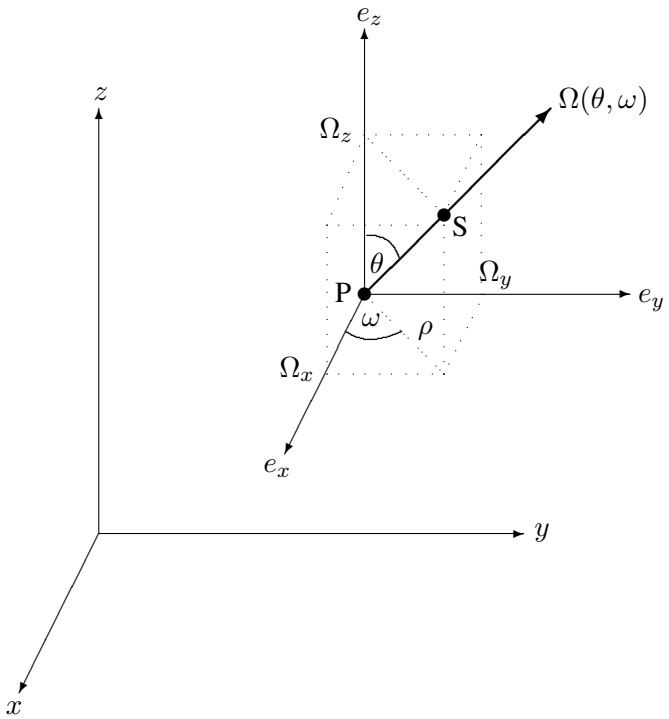


Figure 2.1.: Angle on the unit sphere in terms of Cartesian components.

which the particle is streaming may be thought of as the radius vector of the unit sphere, with its centre at the position of the particle. Point S denoted by a solid circle illustrates a point on the surface of the unit sphere. The angular coordinate system illustrated has an origin at the centre of this unit sphere (the position P of the particle). This centre travels with the particle as it streams along the angle $\Omega(\theta, \omega)$ (which is constant). The angle on which the particle is travelling depends

on the polar angle θ and the azimuthal angle ω, and has a Cartesian component measured along each axis, which are labelled Ω_x, Ω_y and Ω_z in the figure. The angles θ and ω are measured from the pole and in the (e_x, e_y) plane, respectively. The length ρ is defined from the geometry to be $\rho = \sin\theta$ since the radius vector is unity. The Cartesian components may then be written:

$$\begin{aligned} \Omega_x &= \sin\theta\cos(\omega), \\ \Omega_y &= \sin\theta\sin(\omega), \\ \Omega_z &= \cos\theta. \end{aligned} \tag{2.7}$$

Using the expression for Ω_z, the x and y components may be written as:

$$\begin{aligned} \Omega_x &= \left(1-\Omega_z^2\right)^{\frac{1}{2}}\cos\omega, \\ \Omega_y &= \left(1-\Omega_z^2\right)^{\frac{1}{2}}\sin\omega. \end{aligned} \tag{2.8}$$

It is common in radiation transport literature to use the notation $\mu = \Omega_z$. The Cartesian components are then expressed in terms of μ and ω by:

$$\begin{aligned} \Omega_x &= (1-\mu^2)^{\frac{1}{2}}\cos\omega, \\ \Omega_y &= (1-\mu^2)^{\frac{1}{2}}\sin\omega, \\ \Omega_z &= \mu. \end{aligned} \tag{2.9}$$

The final form of the angularly discretised transport equation is obtained by inserting equations (2.6) and (2.9) into equation (2.5) and grouping the removal and scattering operators together.

$$\frac{1}{v}\mathbf{A}_t\frac{\partial\boldsymbol{\Psi}}{\partial t} + \mathbf{A}_x\frac{\partial\boldsymbol{\Psi}}{\partial x} + \mathbf{A}_y\frac{\partial\boldsymbol{\Psi}}{\partial y} + \mathbf{A}_z\frac{\partial\boldsymbol{\Psi}}{\partial z} + \mathbf{H}\boldsymbol{\Psi} = \boldsymbol{S}. \tag{2.10}$$

The variable $\boldsymbol{\Psi} = \boldsymbol{\Psi}(\mathbf{r})$ is the vector of $\mathcal{M}$ angular moments that construct the approximation of the angular flux at position $\mathbf{r}$ in space-time. $\boldsymbol{S} = \boldsymbol{S}(\mathbf{r})$ is the vector of size $\mathcal{M}$ containing the angular discretised external source at position $\mathbf{r}$ in space-time. $\mathbf{A}_t$ is the $\mathcal{M}\times\mathcal{M}$ angular mass matrix, $\mathbf{A}_x$, $\mathbf{A}_y$ and $\mathbf{A}_z$ are the $\mathcal{M}\times\mathcal{M}$ angular Jacobian matrices and $\mathbf{H}$ is the $\mathcal{M}\times\mathcal{M}$ scattering removal operator

matrix. The elements of the above matrices and source vectors are specified as:

$$\begin{aligned}
\mathbf{A}_{t_{ij}} &= \int d\Omega\, G_i(\mathbf{\Omega})\, G_j(\mathbf{\Omega}) = \int d\mu \int d\omega\, G_i(\mu,\omega)\, G_j(\mu,\omega), \\
\mathbf{A}_{x_{ij}} &= \int d\Omega\, \Omega_x\, G_i(\mathbf{\Omega})\, G_j(\mathbf{\Omega}) \\
&= \int d\mu \int d\omega\, (1-\mu^2)^{\frac{1}{2}} \cos(\omega)\, G_i(\mu,\omega)\, G_j(\mu,\omega), \\
\mathbf{A}_{y_{ij}} &= \int d\Omega\, \Omega_y\, G_i(\mathbf{\Omega})\, G_j(\mathbf{\Omega}) \\
&= \int d\mu \int d\omega\, (1-\mu^2)^{\frac{1}{2}} \sin(\omega)\, G_i(\mu,\omega) G_j(\mu,\omega),
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}_{z_{ij}} &= \int d\Omega\, \Omega_z\, G_i(\mathbf{\Omega})\, G_j(\mathbf{\Omega}) \\
&= \int d\mu \int d\omega\, \mu\, G_i(\mu,\omega) G_j(\mu,\omega), \\
\boldsymbol{S}_i &= \int d\Omega\, \boldsymbol{S}(\mathbf{\Omega})\, G_i(\mathbf{\Omega}) = \int d\mu \int d\omega\, \boldsymbol{S}(\mu,\omega)\, G_i(\mu,\omega). \quad (2.11)
\end{aligned}$$

For transport calculations involving anisotropic scattering, it is common practise to expand the probability density function of the scattered angle μ_0 in terms of the orthogonal Legendre polynomials (Lewis and Miller, 1993). Then by applying the addition theorem each Legendre polynomial is expanded in terms of products of two spherical harmonics functions involving the directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$, the directions before and after scattering. This procedure significantly simplifies the calculation of the scattering contribution of $\mathbf{H}$. The full details of the derivation can be found in (Buchan, 2007) but just the results are given here. The elements of matrix $\mathbf{H}$ are given by:

$$\begin{aligned}
\mathbf{H}_{ij} = \int d\Omega\, \sigma_t\, G_i(\mathbf{\Omega})\, G_j(\mathbf{\Omega}) - \Big\{ \sum_{l=0}^{L} \sigma_{sl}(r) \alpha_i^{e,l,0} \alpha_j^{e,l,0} \\
+2 \sum_{l=1}^{L} \sigma_{sl}(r) \sum_{m=1}^{l} \left[\alpha_i^{e,l,m} \alpha_j^{e,l,m} + \alpha_i^{o,l,m} \alpha_j^{o,l,m} \right] \Big\}. \quad (2.12)
\end{aligned}$$

where $\sigma_{sl}(r)$ are the coefficients of the Legendre expansion of the scattering cross section. This expansion has been truncated to order L. The variables α_i are defined

as:

$$\alpha_i^{e,l,m} = \int d\Omega \; G_i(\Omega) Y_{l,m}^e(\Omega),$$
$$\alpha_i^{o,l,m} = \int d\Omega \; G_i(\Omega) Y_{l,m}^o(\Omega), \tag{2.13}$$

where $Y_{l,m}^e$ and $Y_{l,m}^o$ are the real and complex parts of the spherical harmonic function $Y_{l,m}$:

$$\begin{aligned} Y_{l,m}^e &= \sqrt{\frac{(2l+m)(l-m)!}{(l+m)!}} P_l^m cos(m\omega), \\ Y_{l,m}^o &= \sqrt{\frac{(2l+m)(l-m)!}{(l+m)!}} P_l^m sin(m\omega), \end{aligned} \tag{2.14}$$

and P_l^m denote the associated Legendre functions (Lewis and Miller, 1993). This completes the full angular discretisation of the Boltzmann transport equation using an arbitrary angular expansion (2.4). Once the Jacobian matrices have been obtained, the angular discrete transport equation 2.10 may be written in the more compact form:

$$(\mathbf{A} \cdot \nabla + \mathbf{H}(\mathbf{r}))\mathbf{\Psi}(\mathbf{r}) - \mathbf{s}(\mathbf{r}) = \mathbf{0}, \tag{2.15}$$

in which $\mathbf{A}$ is a vector containing the spatially invariant and time- independent $\mathcal{M} \times \mathcal{M}$ angular Jacobian matrices. Therefore, $\mathbf{A} = (\mathbf{A}_t, \mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z)^T$ and is a vector of $\mathcal{M} \times \mathcal{M}$ matrices. Depending on the choice of angular basis one uses to define the components of $\mathbf{A}$, each component may also be rotationally invariant as well as spatially invariant. Note that equation (2.15) is independent of angular scheme. One may use discrete ordinate (S_N), spherical harmonic (P_N) or wavelet (LW_N) methods without having to alter equation (2.15). The angular scheme may be selected simply by evaluating the elements of each Jacobian matrix accordingly. This has the advantage that only one solution algorithm is necessary. Consequently, all the work that follows is completely general and not specific to any particular angular discretisation. For example, to use S_N one would define

$(\mathbf{A}_t, \mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z)^T$ as:

$$\mathbf{A} = \left[\begin{pmatrix} \frac{1}{v_g} & 0 & \dots & 0 \\ 0 & \frac{1}{v_g} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \frac{1}{v_g} \end{pmatrix}, \begin{pmatrix} \mu_1 & 0 & \dots & 0 \\ 0 & \mu_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \mu_{\mathcal{M}} \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} \eta_1 & 0 & \dots & 0 \\ 0 & \eta_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \eta_{\mathcal{M}} \end{pmatrix}, \begin{pmatrix} \xi_1 & 0 & \dots & 0 \\ 0 & \xi_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \xi_{\mathcal{M}} \end{pmatrix} \right]^T . \tag{2.16}$$

in which μ_i, η_i and ξ_i are the x, y and z components of the angle i, that is the direction cosines along each axis that define the abscissae of angle i. For P_N, the angular flux is expanded in terms of the orthonormal spherical harmonics. These are the real part of the solution to Laplace's equation in spherical geometry, and consequently are invariant to rotation. That is to use:

$$\psi(\mathbf{r}, \mathbf{\Omega}) \approx \sum_{l=0}^{L} \sum_{m=-l}^{l} Y_{lm}(\mu, \omega) \Psi_{lm}(\mathbf{r}), \tag{2.17}$$

where μ is the cosine of the polar angle and ω is the azimuthal angle, which the above expression equation (2.17) is exact for $l = \infty$. Thus the approximation made in equation (2.17) is to truncate the expansion at $l = L$ where L is sufficiently large to give an acceptable solution; here L is set to N, the angular expansion order. This closure is the approximation one makes when P_N is used to discretise angular variation, and assumes all moments $\Psi_{lm}(\mathbf{r}) = 0$ for $l > L$. Methods for improving this type of closure by adding a controlled tailing function are currently under development (McClarren, 2011). To use spherical harmonics for describing the angular variation in equation (2.15), one would define element i, j of each

Jacobian matrix as:

$$\begin{aligned}\mathbf{A}_{t_{ij}}, \mathbf{A}_{x_{ij}}, \mathbf{A}_{y_{ij}}, \mathbf{A}_{z_{ij}} = \frac{1}{v_g} \int_{4\pi} d\Omega Y_{lm}(\mu,\omega) Y_{l'm'}(\mu,\omega), \\ \int_{4\pi} d\Omega \Omega_x(\mu,\omega) Y_{lm}(\mu,\omega) Y_{l'm'}(\mu,\omega), \\ \int_{4\pi} d\Omega \Omega_y(\mu,\omega) Y_{lm}(\mu,\omega) Y_{l'm'}(\mu,\omega), \\ \int_{4\pi} d\Omega \Omega_z(\mu,\omega) Y_{lm}(\mu,\omega) Y_{l'm'}(\mu,\omega), \end{aligned} \tag{2.18}$$

and the l, m indices of the spherical harmonic bases Y_{lm} must be related to a row and column address i, j in the angular Jacobian matrix, to fully populate it in a computer code. Typically for P_N, the integrals in equation (2.18) are computed numerically using an appropriate quadrature rule, which involves evaluating $\mathbf{\Omega}(\mu,\omega)$ and $Y_{lm}(\mu,\omega)$ for a very large number of angles. Alternatively, they are obtained analytically, using trigonometric identities. A property of the P_N angular Jacobian matrix that is potentially extremely useful is that they are hierarchical. That is the P_n expansion contains all the $P_{n-2}, P_{n-4}, ..., P_1$ expansions as a subset. Potentially, one can therefore adapt the order of the approximation dynamically during a calculation quite easily, given a suitable metric for the error. This is not true for S_N, where any attempt at adapting the angle requires mapping the angular fluxes to a new quadrature set. This may not be computationally feasible. However, modifications can be made to S_N that render it more suitable to adaption. Such methods produce wavelet like schemes and are akin to performing a finite element discretisation in angle (Buchan, 2007). The purpose of the angular approximation methods discussed in this section is to arrive at the system of equations (2.10). At this point the angular flux profile is fully described by the coefficients $\{\Psi_1(\mathbf{r}), \Psi_2(\mathbf{r}), ..., \Psi_{\mathcal{M}}(\mathbf{r})\}$. This also presents a suitable starting position for developing the techniques for the spatial and time discretisation.

2.4. Space-Time Discretisation

Once the angular part is complete, one is in a position to perform the spatial and temporal discretisation. In the current work, the discontinuous Galerkin method is used on both the spatial and temporal grids. A suitable set of nodal basis func-

tions $\mathbf{N}(\mathbf{r})$ are chosen, where $\mathbf{N}_i(\mathbf{r})$ is a diagonal $\mathcal{M} \times \mathcal{M}$ matrix with the node i basis function along the main diagonal, $\forall i \in \{1, 2, \ldots, \mathcal{N}\}$ where $\mathcal{N}$ is the number of nodes in the finite element mesh. The spatial dependence of $\mathbf{N}_i(\mathbf{r})$ shall be assumed and the matrix represented simply by $\mathbf{N}_i$. Since the present work is concerned with a Galerkin projection, the nodal basis functions are used to weight the residual. Equation (2.15) is multiplied through by $\mathbf{N}_i$. One then substitutes $\Psi(\mathbf{r}) = \sum_j \mathbf{N}_j \mathbf{\Phi}_j$, and $\mathbf{s}(\mathbf{r}) = \sum_j \mathbf{N}_j \mathbf{s}_j$ and integrates over the element domain V_e (the boundary of V_e is Γ_e) to obtain the equation:

$$\int_{V_e} dV(-\mathbf{A} \cdot \nabla \mathbf{N}_i \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_j + \mathbf{N}_i \mathbf{H} \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_j - \sum_j \mathbf{N}_j s_j)$$
$$+ \oint_{\Gamma_e} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{N}_i \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_{IN_j} + \oint_{\Gamma_e} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{N}_i \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_j = \mathbf{0}. \qquad (2.19)$$

Thus each node pair i, j has a matrix block of size $\mathcal{M} \times \mathcal{M}$ associated with it, $\mathcal{M}$ being the number of unknowns of the angular expansion. This gives $\mathcal{M} = N \times (N + 2)$ unknowns for S_N and $\mathcal{M} = (N + 1)^2$ for P_N in three dimensions. The solution $\mathbf{\Phi}_i$ at node $i, \forall i \in \{1, 2, \ldots, \mathcal{N}\}$ is a vector of length $\mathcal{M}$ containing the moments of the flux at that point in space-time. $\mathbf{\Phi}_{IN_i}$ is the boundary condition on node i coming in from either a neighbouring element or the edge of the domain, and $\mathbf{0}$ a vector of length $\mathcal{M}$ containing zeroes. The discontinuities inherent in the solution at each node mean that the spatial derivatives that appear in the transport equation are troublesome. They are removed from the equation when one applies integration by parts and Green's Theorem to the streaming terms. Not only does this remove the derivative from the equation, it produces surface terms that couple the elements in space-time across the mesh. This surface term has been split into an inbound contribution and an outbound contribution on Γ_e. The standard discontinuous Galerkin approach fully up-winds the solution on the incoming boundaries of an element in both space and time. Use of the Riemann method to separate inbound information from outbound information is discussed in section (2.5). The surface integral containing the outbound term is placed on the left hand side of the equation and absorbed into the matrix, forming part of the solution. The inbound integral forms part of the source term on the right hand side of the matrix equation. This completes the spatial and temporal part of the discretisation.

2.5. Riemann Decomposition

The surface integrals in equation (2.19) can be resolved by using a Riemann method to separate the incident and outgoing contributions to the surface term at the boundaries of each element. This section describes how the space-time boundaries are implemented using the finite element Riemann approach. At a surface, the $\mathcal{M}\times\mathcal{M}$ matrix within the integral may be written as $\mathbf{A}_s = (\mathbf{n}\cdot\mathbf{A})$, where $\mathbf{n}$ denotes the outward facing normal to an element edge. The matrix $\mathbf{A}_s$ is decomposed using its eigenstructure:

$$\mathbf{A}_s = \mathbf{L}_s\Lambda_s\mathbf{R}_s. \tag{2.20}$$

The $\mathcal{M}\times\mathcal{M}$ matrix $\mathbf{R}_s$ is formed from the columns of right eigenvectors of $\mathbf{A}_s$, and $\mathbf{L}_s = \mathbf{R}_s^{-1} = \mathbf{R}_s^T$. Λ_s is a $\mathcal{M}\times\mathcal{M}$ diagonal matrix containing its eigenvalues. Considering the surface integral of (2.19), the matrix $\mathbf{R}_s$ (associated with the surface integral decomposition (2.20)) is used to transform the finite element solution vector $\mathbf{\Psi}$ on the boundary into a Riemann space through the mapping:

$$\mathbf{\Psi} = \mathbf{R}_s\mathbf{\Psi}_s, \tag{2.21}$$

where $\mathbf{\Psi}_s$ is a vector of size $\mathcal{M}$ denoting the new Riemann variables. Substituting this expression into the surface integral and using the decomposition in (2.20) results in the following:

$$\oint_{\Gamma_e} d\Gamma\mathbf{N}_i\mathbf{A}_s\mathbf{\Psi} \;=\; \oint_{\Gamma_e} d\Gamma\mathbf{N}_i\mathbf{R}_s\Lambda_s\mathbf{R}_s^{-1}\mathbf{R}_s\mathbf{\Psi}_s\mathbf{R}_s \oint_{\Gamma_e} d\Gamma\mathbf{N}_i\Lambda_s\mathbf{\Psi}_s. \tag{2.22}$$

In this form, the surface integral is expressed in terms of the Riemann variables which are decoupled due to Λ_s being diagonal. Furthermore, the direction of information of each Riemann variable can be determined by the sign of its associated eigenvalue of Λ_s. That is, the k^{th} element of the Riemann variable vector $\mathbf{\Psi}_{s_k}, \forall k \in \{1, 2, \ldots \mathcal{M}\}$, is found to have its information entering an element if the eigenvalue $\Lambda_{s_{k,k}}$ is negative. Otherwise, for positive values, the information of the variable will be outgoing through the surface. This allows the last integral in equation (2.22) to be re-written as the sum of the incoming ($\mathbf{\Psi}_{s_{in}}$) and outgoing ($\mathbf{\Psi}_{s_{out}}$) information,

$$\oint_{\Gamma_e} d\Gamma\mathbf{N}_i\Lambda_s\mathbf{\Psi}_s = \oint_{\Gamma_{in}} d\Gamma\mathbf{N}_i\Lambda_s\mathbf{\Psi}_{s_{in}} + \oint_{\Gamma_{out}} d\Gamma\mathbf{N}_i\Lambda_s\mathbf{\Psi}_{s_{out}}, \tag{2.23}$$

where the integrals $\oint_{\Gamma in}$ and $\oint_{\Gamma out}$ represent integrals across the surface for incoming and outgoing information. From this the incoming information can be set to zero in order to resolve a bare surface. In the present work, only structured orthogonal grids are used. This allows the components of $\mathbf{A}$, each of which is also an $\mathcal{M} \times \mathcal{M}$ matrix, to be treated independently. That is to express each Jacobian $\mathbf{A}_t, \mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$ as an independent eigenvalue problem:

$$\mathbf{A}_x = \mathbf{L}_x \mathbf{\Lambda}_x \mathbf{R}_x, \quad \mathbf{A}_y = \mathbf{L}_y \mathbf{\Lambda}_y \mathbf{R}_y, \quad \mathbf{A}_z = \mathbf{L}_z \mathbf{\Lambda}_z \mathbf{R}_z, \quad \mathbf{A}_t = \mathbf{L}_t \mathbf{\Lambda}_t \mathbf{R}_t, \tag{2.24}$$

where $\mathbf{L}_x, \mathbf{L}_y, \mathbf{L}_z$ are the non-diagonal matrices of left eigenvectors, $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ the non-diagonal matrices of right eigenvectors and $\mathbf{\Lambda}_x, \mathbf{\Lambda}_y, \mathbf{\Lambda}_z$ the diagonal matrices of eigenvalues associated with the Riemann decomposition of $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$ respectively. Positive eigenvalues correspond with outbound information on the element boundaries and negative eigenvalues correspond with incoming information. For orthogonal angular basis functions, such as those considered here, the matrix $\mathbf{A}_t$ is diagonal, therefore $\mathbf{\Lambda}_t = \mathbf{A}_t$ and $\mathbf{L}_t = \mathbf{R}_t = \mathbf{I}$. Once the eigenstructure of each Jacobian is obtained, the components of each eigenvalue matrix $\mathbf{\Lambda}_x, \mathbf{\Lambda}_y, \mathbf{\Lambda}_z$ are sorted by sign in the Riemann space where they are diagonal. This involves constructing two diagonal sorting matrices for each axis to decouple the directions into separate inbound and outbound modes:

$$\mathbf{\Lambda}^+_{x_{ii}} = \begin{cases} \mathbf{\Lambda}_{x_{ii}} & \mathbf{\Lambda}_{x_{ii}} \cdot n_x \geq 0; \\ 0 & \mathbf{\Lambda}_{x_{ii}} \cdot n_x < 0. \end{cases}, \quad \mathbf{\Lambda}^-_{x_{ii}} = \begin{cases} \mathbf{\Lambda}_{x_{ii}} & \mathbf{\Lambda}_{x_{ii}} \cdot n_x < 0; \\ 0 & \mathbf{\Lambda}_{x_{ii}} \cdot n_x \geq 0. \end{cases} \tag{2.25}$$

$$\mathbf{\Lambda}^+_{y_{ii}} = \begin{cases} \mathbf{\Lambda}_{y_{ii}} & \mathbf{\Lambda}_{y_{ii}} \cdot n_y \geq 0; \\ 0 & \mathbf{\Lambda}_{y_{ii}} \cdot n_y < 0. \end{cases}, \quad \mathbf{\Lambda}^-_{y_{ii}} = \begin{cases} \mathbf{\Lambda}_{y_{ii}} & \mathbf{\Lambda}_{y_{ii}} \cdot n_y < 0; \\ 0 & \mathbf{\Lambda}_{y_{ii}} \cdot n_y \geq 0. \end{cases} \tag{2.26}$$

$$\mathbf{\Lambda}^+_{z_{ii}} = \begin{cases} \mathbf{\Lambda}_{z_{ii}} & \mathbf{\Lambda}_{z_{ii}} \cdot n_z \geq 0; \\ 0 & \mathbf{\Lambda}_{z_{ii}} \cdot n_z < 0. \end{cases}, \quad \mathbf{\Lambda}^-_{z_{ii}} = \begin{cases} \mathbf{\Lambda}_{z_{ii}} & \mathbf{\Lambda}_{z_{ii}} \cdot n_z < 0; \\ 0 & \mathbf{\Lambda}_{z_{ii}} \cdot n_z \geq 0. \end{cases} \tag{2.27}$$

where n_x, n_y, n_z and n_t define the outward unit normal to an element face for direction x, y, z and t respectively. For example, orthogonal quadrilateral elements aligned with the coordinate axes in Cartesian geometry have $n_x = -1$ for faces whose outward normal points anti-parallel to the x-axis, and $n_x = +1$ for faces whose outward normal points parallel to the x-axis. Thus $\mathbf{\Lambda}^+_x$ is associated in Rie-

mann space with outgoing information along the x direction and $\mathbf{\Lambda}_x^-$ is associated in Riemann space with incoming information along the x direction. The other axes are treated likewise. The Riemann approach allows each axis to be treated as an independent eigenvalue problem. Once formed, the sorting matrices are mapped back in the associative transforms:

$$\begin{aligned}
\mathbf{A}_x^+ &= \mathbf{L}_x\mathbf{\Lambda}_x^+\mathbf{R}_x, \quad \mathbf{A}_x^- = \mathbf{L}_x\mathbf{\Lambda}_x^-\mathbf{R}_x, \\
\mathbf{A}_y^+ &= \mathbf{L}_y\mathbf{\Lambda}_y^+\mathbf{R}_y, \quad \mathbf{A}_y^- = \mathbf{L}_y\mathbf{\Lambda}_y^-\mathbf{R}_y, \\
\mathbf{A}_z^+ &= \mathbf{L}_z\mathbf{\Lambda}_z^+\mathbf{R}_z, \quad \mathbf{A}_z^- = \mathbf{L}_z\mathbf{\Lambda}_z^-\mathbf{R}_z, \\
\mathbf{A}_t^+ &= \mathbf{L}_t\mathbf{\Lambda}_t^+\mathbf{R}_t, \quad \mathbf{A}_t^- = \mathbf{L}_t\mathbf{\Lambda}_t^-\mathbf{R}_t.
\end{aligned} \tag{2.28}$$

The two surface integrals in equation (2.19) are then split up into a contribution for each axis. The integral containing all the inbound information is written:

$$\begin{aligned}
\oint_{\Gamma_{ein}} d\Gamma \mathbf{n}\cdot\mathbf{AN}_i\Psi_{IN} = &\oint_{\Gamma_{x_{IN}}} d\Gamma n_x\mathbf{A}_x^-\Psi_{x_{IN}} + \oint_{\Gamma_{x_{IN}}} d\Gamma n_y\mathbf{A}_y^-\Psi_{y_{IN}} \\
&+ \oint_{\Gamma_{x_{IN}}} d\Gamma n_z\mathbf{A}_z^-\Psi_{z_{IN}} + \oint_{\Gamma_{x_{IN}}} d\Gamma n_t\mathbf{A}_t^-\Psi_{t_{IN}},
\end{aligned} \tag{2.29}$$

and the integral containing all the outbound information written

$$\begin{aligned}
\oint_{\Gamma_{eout}} d\Gamma \mathbf{n}\cdot\mathbf{AN}_i\Psi = &\oint_{\Gamma_{x_{out}}} d\Gamma n_x\mathbf{A}_x^+\Psi_{x_{out}} + \oint_{\Gamma_{y_{out}}} d\Gamma n_y\mathbf{A}_y^+\Psi_{y_{out}} \\
&+ \oint_{\Gamma_{z_{out}}} d\Gamma n_z\mathbf{A}_z^+\Psi_{z_{out}} + \oint_{\Gamma_{t_{out}}} d\Gamma n_t\mathbf{A}_t^+\Psi_{t_{out}}.
\end{aligned} \tag{2.30}$$

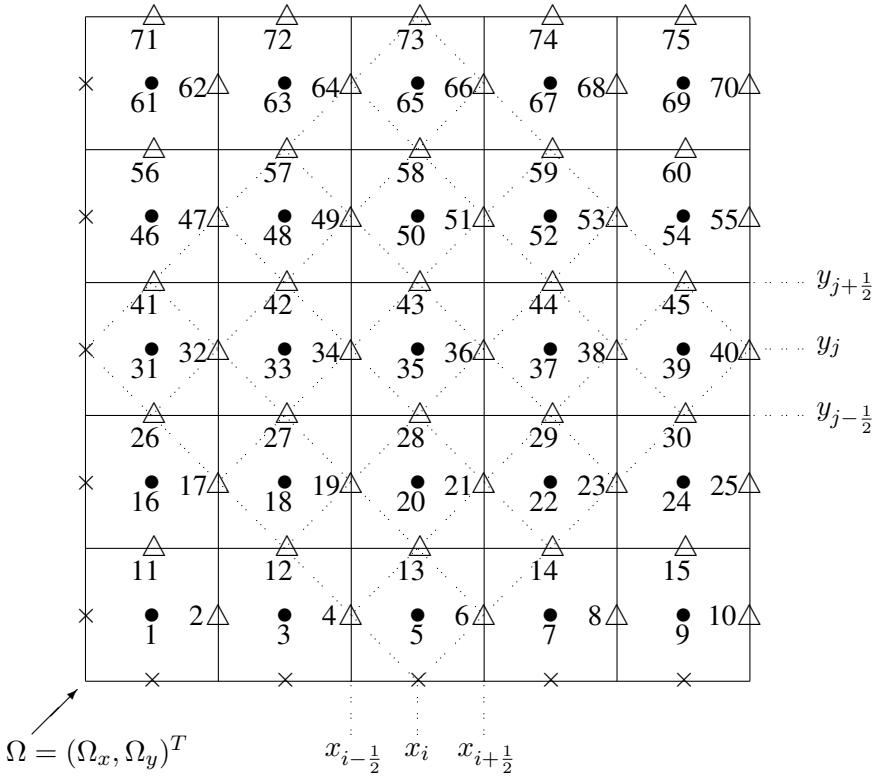
Transform (2.28) produces two new angular Jacobians for each element face. $\mathbf{A}_x^+$ is used to angularly discretise the surface integral containing outbound information in the x direction. Thus it is placed in the integral $\Gamma_{x_{out}}$ in equation (2.30). $\mathbf{A}_x^-$ is used to angularly discretise the surface integral containing the incoming information in the x direction. Thus it is placed in the integral $\Gamma_{x_{IN}}$ in equation (2.29). The y, z and t surface terms are treated likewise, completely independently. Once formed, the surface blocks are mapped into matrix blocks and placed into the discretised matrix equation. The surface block containing outbound information is added on to the left hand side of the matrix equation so the unknown outbound information is absorbed into the solution vector. The surface block containing in-

bound information is added on to the source vector on the right hand side since it is known from the downwind solution. This assumes one is applying a sweep type solver, and the order in which the elements must be swept is known. Unstructured meshes that are cyclic require a more sophisticated solution method. The finite element Riemann method allows the angular discretisation to be chosen simply by evaluating the vector $(\mathbf{A}_t, \mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z)^T$ appropriately without any distinctions having to be made between the different schemes anywhere in the code, not even inside the solution algorithm itself. This method provides spatial coupling of the elements and places the numerical schemes that follow within a Riemann framework. It is similar to the Riemann-Rudyard approach (Pain et al., 2006a) where each coordinate is also described separately, replacing a multi-dimensional system with a set of independent one-dimensional waves. Other strategies for implementing boundary conditions in non-diagonal systems include Marshak boundary conditions (Stacey, 2007; Andersen, 1997), such as those implemented in the spherical harmonics finite element code *EVENT* . These tend to require the evaluation of half-range integrals, a huge computational overhead on arbitrary meshes. It is believed the Riemann method implemented here to be a considerable improvement in computational efficiency.

2.6. Mesh-Angle Sweep

All schemes in the present work are implemented on an orthogonal structured mesh in Cartesian geometry. Advantage of the regular mesh is taken in the solution schemes, using a fixed sweeping pattern. The sweeping pattern used is that of the well-known diamond method (Lewis and Miller, 1993), and is illustrated in figure (2.2). This diagram is based on that found on page 170 of (Lewis and Miller, 1993) and illustrates a 2-dimensional computational mesh in Cartesian (x, y) planar geometry. The mesh in the figure contains 5 cells in each direction, with the edges (vertices) of cell (i, j) at positions $(x_{i\pm\frac{1}{2}}, y_{j\pm\frac{1}{2}})$ and the cell centre at position (x_i, y_j). A sweep may be defined as a list of all the cells on the mesh in the order they must be solved. In the example shown, particles travelling in direction $\Omega = (\Omega_x, \Omega_y)^T$ from left-to-right and bottom-to-top enter the problem along the left and lower edges of the mesh. For this particular angle, the left and lower edges of the mesh can contain only inbound information. Particles that are outbound through these surfaces are travelling in directions other than Ω, and will have at least one negative Cartesian component of their angle. The known boundary val-

ues on the left and lower edges of the mesh are denoted by the symbol $\times$, and are prescribed at the mid-points of the cell boundaries in the positions shown in the figure. The sweep for this angle must therefore begin in the lower-left corner cell, as this is the only cell that is not coupled to another cell for information travelling on this angle. The sweep proceeds from left-to-right and bottom-to-top, solving the cells in the order indicated by the numbers. In the spatial finite difference (di-



● Solution calculated at cell centre
△Interface value to advance solution
×Incoming boundary condition on edge of domain

Figure 2.2.: Diamond-Difference mesh sweeping scheme in 2D.

amond difference) method, the computational solution is obtained at the centre of each cell on the structured grid. The location of each solution point is denoted by a solid circle in the figure, and numbered in the order it is solved. Once the cell-centre value has been obtained from the discrete equation (formed via a volume averaging of the flux in each cell), the upwind information which is unknown is eliminated from the equations. This is done using a balance relation. The balance

relation expresses the flux on each outbound edge in terms of the cell-centre solution and the incoming information from the previous cells on the sweep. The outbound edge fluxes are denoted by a triangle in the figure. It is these fluxes that advance the sweep in the chosen direction, by coupling cell (i, j) to the downwind cells $(i + 1, j)$ and $(i, j + 1)$. Thus a data front moves across the mesh to the opposite corner from where the sweep was started. Of course, the definition of downwind (and therefore which quantities in the equation are unknown and need to be eliminated with a balance equation) changes depending on which angle is being solved. The sweep order that is illustrated (referred to as the sweep graph) solves for all angles of particle travel that have an x-component $\Omega_x > 0$ and a y-component $\Omega_y > 0$. For other angles, the cells must be swept in a different order, with their sweep origin in a different corner, and with a different definition of which direction (and therefore which cell) is upwind. Dotted diagonal lines have been superimposed on the figure to illustrate the reason why the scheme is referred to as the diamond scheme; the edge fluxes at the cell interfaces are defined at the mid points. It is these values that advance the sweep in the chosen direction. In this example, a constant mesh spacing is used in both the x and y directions. This is not a requirement for the methods in the current work, and an arbitrary spacing may be used without any added complexity of implementation. For cell (i, j) in the grid, the width in the x direction is defined as $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and the width in the y direction defined as $\Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$.

2.7. Source Iteration

In general, the source term in the discrete neutron transport equation contains a fission source and a scattering source. Particles of different energies have different contributions to these sources, complicating the solution of the multi-group equations. In the case of the scattering source, particles are removed from one energy and sourced into another by the out-of-group scattering process. For example, in a moderating material high energy neutrons will be slowed, removing them from the fast group and placing them in a lower energy group. This is referred to as *down-scatter*. Conversely, neutrons travelling with a low energy can be scattered up in energy via interaction with a higher-energy nucleus. This is known as *up-scatter*, and gives rise to a neutron source in higher energy groups. Similarly, particles can scatter in angle and there will be additional contributions to the scattering source due to this. Thus each energy group and each angle has a neutron source (and a re-

moval) due to the scattering processes coupling the different energy levels and the scattering processes coupling the different angles. This coupling complicates the solution strategy, as the flux in any given group or angle depends on a source that has contributions from all energy groups and angles, including those that have yet to be updated by the algorithm. One typically uses an iterative procedure, which allows each energy group to be treated as an independent fixed source problem. The iteration continues until the solution in each group has converged. This is done for each energy group in turn and is known as *source iteration* (Azmy and Sartori, 2010; Lewis and Miller, 1993). The procedure is described in detail elsewhere (Azmy and Sartori, 2010), and only a brief summary is given here. It may be written for the one-dimensional S_N equations with G discrete energy groups and $\mathcal{M}$ angles as:

$$\begin{aligned}\left(\frac{\partial \mu_m}{\partial x} + \sigma_{t_g}(x)\right)\psi^k_{m,g}(x) &= q_{ext_{m,g}}(x) \\ &+ \sum_{g'=1}^{G}\sum_{m'=1}^{\mathcal{M}} \sigma_{s_{m'\to m,g'\to g}}(x)\psi^{k-1}_{m',g'}(x) \\ &+ \sum_{m'=1}^{\mathcal{M}} \sigma_{f_g}(x) w_{m'}\psi^{k-1}_{m',g}(x), \qquad (2.31)\end{aligned}$$

in which μ_m is the direction cosine of angle m and g the energy group. The total cross-section for group g at position x is denoted by $\sigma_{t_g}(x)$. The cross-section for the scattering of particles from energy g' into energy g and from angle m' into angle m is denoted by $\sigma_{s_{m'\to m,g'\to g}}(x)$. The fission cross-section is denoted by $\sigma_{f_g}(x)$ and the S_N quadrature weight for angle m by w_m. External sources are referred to as $q_{ext_{m,g}}(x)$ and the solution at the k^{th} iteration given by $\psi^k_{m,g}(x)$. One typically loops over the number of energy groups in order of descending energy (increasing g), at each source iteration until the solution in every group has converged. The source-iteration procedure is known to converge (Azmy and Sartori, 2010); however there are many problems for which convergence is slow. Improvements to the iteration count may be obtained by splitting the scattering source into a within-group ($g' = g$) component, a down-scatter component ($g' > g$) and an up-scatter component ($g' < g$). This results in a formulation that is similar to the

Gauss- Seidel method (Kreyszig, 1993), and may be written for equation (2.31) as:

$$\left(\frac{\partial \mu_m}{\partial x} + \sigma_{t_g}(x) - \sigma_{s_{m\longrightarrow m,g\longrightarrow g}}(x)\right)\psi^k_{m,g}(x) - \sum_{g'=1}^{g-1}\sigma_{s_{m,g'\rightarrow g}}(x)\psi^k_{m,g'}(x)$$
$$= q_{ext_{m,g}}(x) + \sum_{g'=g+1}^{G}\sigma_{s_{m,g'\rightarrow g}}(x)\psi^{k-1}_{m,g'}(x)$$
$$+ \sum_{m'=1}^{\mathcal{M}}\left(\sigma_{f_g}w_{m'}(x) + \sigma_{s_{m'\longrightarrow m,g}}(x)\right)\psi^{k-1}_{m',g}(x). \quad (2.32)$$

In addition to the scattering source, which requires the flux solution for all energy groups, the fission source is similarly unknown until the multi-group scattering source iteration has converged. This is because neutrons of different energies have different contributions to the fission source. Thus a second ”outer” inverse power iteration on the fission source is usually performed which contains a nested ”inner” multi-group scattering source iteration. This procedure is illustrated in figure (B.1) in appendix (B). It converges the scattering source for a given energy group so the contribution of that group to the fission source can be updated until the number of fissions has converged.

2.8. Discontinuous Methods

A number of discontinuous Galerkin (DG) methods for radiation transport have been implemented in a one-dimensional code and these are discussed in detail in the present section. These methods represent different discretisation schemes based on DG and they differ in how the solution is up-winded on the element boundaries. The boundary term couples neighbouring elements as shown in section (2.4), and how the information in this term is treated is a potential avenue to improving the fidelity of the computational solution. How the boundary term is treated may also have implications for the diffusion limit of a scheme. This should be considered as further work, though has not been done here. Five implementations of the DG method are explored, using different up-winding strategies. These include the standard discontinuous Galerkin scheme itself, implemented without any modifications or enhancement, and four other methods based on this. The four other methods include the Upwind Average DG (UWDG) method which is the standard DG method with averaging of the nodal fluxes on the element bound-

ary rather than the full upwind value, two Petrov-Galerkin schemes and the DG scheme with sub-grid scale bubble functions used to enhance the discretisation. Different methods of weighting these bubble functions are considered. Although lumped linear discontinuous methods are not considered, the schemes that are considered are related to this, because they introduce additional upwinding in the finite element stencil. The Petrov- Galerkin schemes introduce artificial dissipation into the discretisation. This is done by adding on a second order term which appears in the surface integral. In one method, the second order surface terms are included and in the other they are omitted which simplifies the discretisation. The performance of the schemes is demonstrated using Taylor analysis and numerical testing. The section concludes by selecting one of the methods for further study. This involves extending it to multi-dimensions and developing it further in the chapters that follow.

2.8.1. Standard Discontinuous Galerkin

The standard discontinuous Galerkin scheme is applied to an advection problem in one-dimension. In this case, $\mathbf{r}$ becomes x and the equation is written, for a neutral particle travelling in the direction μ as:

$$\mu\frac{\partial\psi(x)}{\partial x}+\sigma(x)\psi(x)-q(x)=0, \tag{2.33}$$

in which $\psi(x)$ is the particle flux at position x, $\sigma(x)$ the scattering-removal operator at position x and $q(x)$ the source term at position x. Assuming the one-dimensional solution domain has been divided into I intervals (or elements), and the cross-section may be regarded as constant throughout a given interval, we may write $\sigma(x)$ as $\sigma_i, \forall i \in \{1,2,\ldots,I\}$. That is to regard σ_i as an element-centred quantity. This simplifies the discretisation of the equations and their integration[2]. Equation (2.33) is solved for $\psi(x)$ based on an estimate of $q(x)$, with an outer iteration loop in the code to converge $q(x)$ in problems where it contains contributions from scattering and/or fission terms. The standard DG method involves substitution of trial functions for these two quantities $\overline{\psi}(x)\approx\psi(x)$, and $\overline{q}(x)\approx q(x)$:

$$\overline{\psi}(x)=\sum_{j=1}^{n}\mathbf{N}_j(x)\psi_j, \qquad \overline{q}(x)=\sum_{j=1}^{n}\mathbf{N}_j(x)q_j, \tag{2.34}$$

[2] Although this treatment can result in large discontinuities in the flux gradient at material boundaries

where in one spatial dimension n=2 and the two shape functions are typically defined as:

$$\mathbf{N}_1(x) = \frac{x_{i+\frac{1}{2}} - x}{\Delta x_i}, \qquad \mathbf{N}_2(x) = \frac{x - x_{i-\frac{1}{2}}}{\Delta x_i}. \tag{2.35}$$

The trial functions in equation (2.34) approximate $\psi(x)$ and $q(x)$ are in equation (2.33), which may therefore be written:

$$\mu \frac{\partial \overline{\psi}(x)}{\partial x} + \sigma_e \overline{\psi}(x) - \overline{q}(x) = \mathscr{R}, \tag{2.36}$$

where $\mathscr{R}$ is known as the equation *residual*. Setting $\mathscr{R} = 0$ does not bring one any closer to a method of solution for equation (2.33). Therefore, in order to proceed it is imposed that $\mathscr{R}$ be arbitrarily small in some sense. To achieve this, a suitable weight function $w(x)$ is used to distribute the error over the domain of the integration, such that it has an average of zero. In the case of discontinuous finite elements, integration is performed over an element. In the case of continuous finite element methods, this integration is performed over the whole domain. The volume of an element is denoted as V_E. In the case of Galerkin weighting, the shape functions defined in equation (2.35) are used as these weights. That is to use $w(x) = \mathbf{N}_k(x)$, in which case one may write:

$$\int_{V_E} dV \mathbf{N}_k(x) \mathscr{R} = 0, \forall k \in \{1, 2, \ldots, , n\}. \tag{2.37}$$

This generates n simultaneous equations in n unknowns. If a fission source is present this is iterated on until the solution converges. The standard DG scheme without a fission source does not require an iterative procedure as there is no dependence on downstream information. Substituting equation (2.36) into equation (2.37) for $\mathscr{R}$ we obtain:

$$\int_{V_E} dV \mathbf{N}_k(x) \left(\mu \frac{\partial \overline{\psi}(x)}{\partial x} + \sigma_i \overline{\psi}(x) - \overline{q}(x) \right) = 0. \tag{2.38}$$

The first term in brackets contains a spatial derivative of the flux function $\overline{\psi}(x)$. This derivative is problematic because we require $\overline{\psi}(x)$ to be discontinuous at element boundaries. To remove it, one first applies the chain rule to the advection term:

$$\mathbf{N}_k(x) \mu \frac{\partial \overline{\psi}(x)}{\partial x} = \mu \frac{\partial}{\partial x} \left(\mathbf{N}_k(x) \overline{\psi}(x) \right) - \overline{\psi}(x) \mu \frac{\partial \mathbf{N}_k(x)}{\partial x}, \tag{2.39}$$

and subsequently, one integrates the advection term and applies Green's Theorem to convert the integration domain from a volume to a surface interval Γ_E:

$$\begin{aligned}\int_{V_E} dV \mathbf{N}_k(x)\mu\frac{\partial\overline{\psi}(x)}{\partial x} &= \int_{V_E} dV\left(\mu\frac{\partial\left(\mathbf{N}_k(x)\overline{\psi}(x)\right)}{\partial x} - \overline{\psi}(x)\mu\frac{\partial\mathbf{N}_k(x)}{\partial x}\right)\\ &= \oint_{\Gamma_E} d\Gamma n_x\cdot\mu\mathbf{N}_k(x)\overline{\psi}(x)\\ &- \int_{V_E} dV\overline{\psi}(x)\mu\frac{\partial\mathbf{N}_k(x)}{\partial x}. \end{aligned} \tag{2.40}$$

Inserting the integration limits over the element volume one obtains:

$$\begin{aligned}\int_{V_E} dV \mathbf{N}_k(x)\mu\frac{\partial\overline{\psi}(x)}{\partial x} &= \oint_{\Gamma_E} d\Gamma n_x\cdot\mu\mathbf{N}_k(x)\overline{\psi}(x)\\ &- \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\overline{\psi}(x)\mu\frac{\partial\mathbf{N}_k(x)}{\partial x}, \end{aligned} \tag{2.41}$$

in which n_x is the outward normal to the element surface in the direction x, i.e. $n_x\cdot\mu$ = -1 for incoming flux. One then substitutes equation (2.41) into equation (2.38) to obtain, for element i, the equation:

$$\begin{aligned}-\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\overline{\psi}(x)\mu\frac{\partial\mathbf{N}_k(x)}{\partial x} &+ \oint_{\Gamma_E} d\Gamma n_x\cdot\mu\mathbf{N}_k(x)\overline{\psi}(x)\\ &+ \sigma_i\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\mathbf{N}_k(x)\overline{\psi}(x)\\ &= \int_{V_E} dx\mathbf{N}_k(x)\overline{q}(x). \end{aligned} \tag{2.42}$$

This eliminates the problematic $\overline{\psi}(x)$ derivative. In the case of no sources, that is the problem is driven entirely by a flux travelling in direction μ entering the

domain at the edge, one may write:

$$-\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\overline{\psi}(x)\frac{\partial \mathbf{N}_k(x)}{\partial x} + \oint_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_k(x)\overline{\psi}(x) + \sigma_i \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\mathbf{N}_k(x)\overline{\psi}(x) = 0. \tag{2.43}$$

Substituting the trial functions (2.34) into the integrals internal to the element one obtains:

$$-\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\mathbf{N}_{x_i}(x)\left(\sum_{j=1}^{n}\mathbf{N}_j(x)\psi_j\right) + \oint_{\Gamma_e} d\Gamma n_x \cdot \mu \mathbf{N}_k \psi_{IN} + \oint_{\Gamma_e} d\Gamma n_x \cdot \mu \mathbf{N}_k \psi_{OUT} + \sigma_i \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx\mathbf{N}_k(x)\left(\sum_{j=1}^{n}\mathbf{N}_j(x)\psi_j\right) = 0, \tag{2.44}$$

in which the node k basis function derivative has been abbreviated as $\mathbf{N}_{x_k}(x)$. The integral that is along the element surface is problematic because it is along the discontinuity in $\overline{\psi}(x)$, and thus the trial function relations (2.34) do not apply. Instead, this surface integral has been split into an inbound and an outbound contribution which have been represented in the integrand with ψ_{IN} and ψ_{OUT} respectively. In the standard DG scheme, the values of the solution on the boundary are set to the value upwind of the corresponding node. Of course, the definition of upwind changes according to the direction in which the particles are travelling, and the convention that is commonly used is that $\mu > 0$ corresponds to particles travelling (being advected) from left to right, and so upwind information is to the left. $\mu < 0$ therefore corresponds to information travelling from right to left, in which case upwind information is on the right. To implement a practical solution scheme for equation (2.44), a matrix representation must first be formulated. Section (2.8.2) describes how this is achieved.

2.8.2. Matrix Representation

Section (2.8.1) derived a discontinuous Galerkin representation of the steady-state advection equation in one spatial dimension (equation (2.44)). By setting $\mathbf{N}_k(x)$ = $\mathbf{N}_1(x)$ and $\mathbf{N}_k(x)$ = $\mathbf{N}_2(x)$ in turn, two simultaneous equations in two unknowns are generated. We also use the fact that, from equation 2.35

$$\mathbf{N}_1(x = x_{i-\frac{1}{2}}) = 1, \tag{2.45}$$
$$\mathbf{N}_1(x = x_{i+\frac{1}{2}}) = 0, \tag{2.46}$$
$$\mathbf{N}_2(x = x_{i-\frac{1}{2}}) = 0, \tag{2.47}$$
$$\mathbf{N}_2(x = x_{i+\frac{1}{2}}) = 1. \tag{2.48}$$

Finally, we upwind ψ_{IN} and ψ_{OUT} to their upstream values, according to the sign of μ. This is illustrated in figure (2.3) for mesh cell i, in which $\mu > 0$ is the

Figure 2.3.: Discontinuities across an element

left-right sweep and $\mu < 0$ is the right-left sweep. On the $\mu > 0$ sweep, ψ_{IN}= ψ_2^{i-1} and ψ_{OUT}=ψ_2^i since the incoming boundary is at x=$x_{i-\frac{1}{2}}$ and the outgoing boundary is at x=$x_{i+\frac{1}{2}}$. On the $\mu < 0$ sweep, ψ_{IN}=ψ_1^{i+1} and ψ_{OUT}=ψ_1^i since the incoming boundary is at x=$x_{i+\frac{1}{2}}$ and the outgoing boundary is at x=$x_{i-\frac{1}{2}}$. In figure (2.3), a solid circle denotes a nodal flux value that is upwind of element i, and an empty circle denotes a nodal flux value internal to element i. Substituting each shape function in turn, and upwinding the equation at the element nodes, the simultaneous equations:

$$\int_{V_E} dV \left(-\mathbf{N}_{x_1}(x) + \sigma_i \mathbf{N}_1(x)\right) \sum_j \mathbf{N}_j(x)\psi_j = -\oint_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_1(x)\psi_2^{i-1},$$
$$\int_{V_E} dV \left(-\mathbf{N}_{x_2}(x) + \sigma_i \mathbf{N}_2(x)\right) \sum_j \mathbf{N}_j(x)\psi_j = -\oint_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_2(x)\psi_2^{i}, \tag{2.49}$$

are produced for particles travelling in the direction $\mu > 0$, and:

$$\int_{V_E} dV \left(-\mathbf{N}_{x_1}(x) + \sigma_i \mathbf{N}_1(x)\right) \sum_j \mathbf{N}_j(x)\psi_j = -\oint_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_1(x)\psi_1^i,$$
$$\int_{V_E} dV \left(-\mathbf{N}_{x_2}(x) + \sigma_i \mathbf{N}_2(x)\right) \sum_j \mathbf{N}_j(x)\psi_j = -\oint_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_2(x)\psi_1^{i+1}, \quad (2.50)$$

are produced for particles that are travelling in the $\mu < 0$ direction. Integration of these two systems of simultaneous equations is based on Gaussian quadrature; the procedure is outlined in (Merton et al., 2008). Completing this numerical integration converts each pair of simultaneous equations into two matrix equations, which we write for $\mu < 0$ as:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} \psi_{Prev} \\ -\psi_2 \end{bmatrix}, \quad (2.51)$$

in which the coefficient $a_{i,j}$ in the above matrix is defined, from the equation (2.50) to be:

$$a_{i,j} = \int_V dV \left(-\frac{\partial \mathbf{N}_i(x)}{\partial x}\mathbf{N}_j(x) + \sigma(x)\mathbf{N}_i(x)\mathbf{N}_j(x)\right), \quad (2.52)$$

and we have substituted for ψ_{IN} and ψ_{OUT} ψ_{Prev} and ψ_2 respectively, in which ψ_{Prev} is the outbound flux from mesh element on the left, the previous mesh element on the sweep. This fully up-winds the scheme in the $\mu > 0$ direction. In the substitution for ψ_{OUT}, ψ_2 is the outbound flux from the mesh element being solved for, i.e. it is part of the solution vector and must be taken over to the left-hand side of the matrix where it is absorbed into the solution:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} + 1 \end{pmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} \psi_{Prev} \\ 0 \end{bmatrix}. \quad (2.53)$$

With the equation in this form, it may be convenient to define the matrix:

$$L = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad (2.54)$$

which we use later to generalise the appearance of the equation. Similarly for $\mu < 0$, we obtain the matrix equation:

$$\begin{pmatrix} a_{11}+1 & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -\psi_{Prev} \end{bmatrix}, \tag{2.55}$$

in which we substitute for ψ_{IN} and ψ_{OUT} ψ_1 and ψ_{Prev} respectively, in which ψ_{Prev} is the outbound flux from mesh element to the right, the previous mesh element on the sweep. This fully up-winds the scheme in the $\mu < 0$ direction. In the substitution for ψ_{OUT}, ψ_1 has been taken over to the left-hand side of the matrix where it is absorbed into the solution. With the equation in this form, it may be convenient to define the matrix:

$$L = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.56}$$

Using our definitions of L which change according to sweep, we are able to generalise the matrix equations into the form:

$$\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + L \right) \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \mathbf{b}, \tag{2.57}$$

which is of the form A**x**=**b**, susceptible to inverting A by a direct method. The definition of the column vector $\mathbf{b}$ also varies according to sweep:

$$\mathbf{b} = \begin{cases} (\psi_{Prev} \quad 0)^T, \mu > 0; \\ (0 \quad \psi_{Prev})^T, \mu < 0, \end{cases} \tag{2.58}$$

in which ψ_{Prev} is the angular flux flowing out from the previous mesh cell on the sweep. Section (2.8.1) shows that the standard DG scheme is working well, comparing favourably with an the analytic solution. Section (2.8.3) describes an alternative scheme for upwinding the flux drive at the element nodes.

2.8.3. Upwind Average Approach

The standard DG method described in section (2.8.1) is up to third order accurate in terms of the spatial truncation, as analysis of the Taylor expansion shows (this is demonstrated in section (2.9). However, being fully up-winded it introduces a high degree of numerical dissipation. The upwind average DG method is com-

pared to the standard DG method in figure (2.4). This method reduces the amount

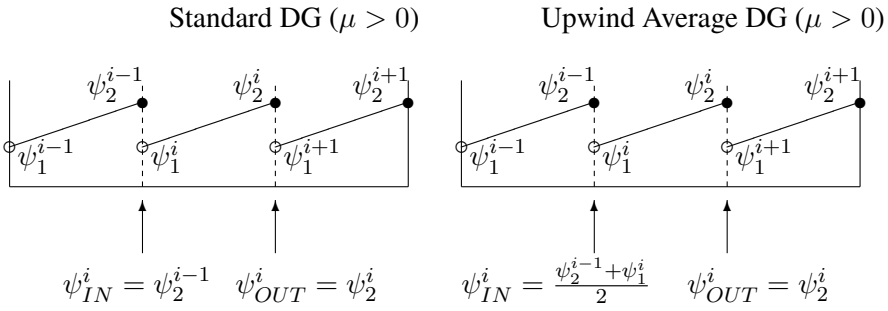


Figure 2.4.: Upwind Average DG and standard DG Method

of upwinding on the incoming boundary. This therefore reduces the dissipative properties of the standard scheme. In the upwind average approach, a mean average of the upwind flux, and the nodal value internal to the mesh cell being solved for, is taken as opposed to fully upwinding the surface value. This is done only for the inbound surface term and not for the outbound surface term. The differences between the two schemes at the nodes are illustrated in figure (2.4). In this illustration, a solid circle denotes a nodal flux that is upwind of a flux being solved for. The diagram is relative to mesh element i (the subject of solution) and is in respect of the $\mu > 0$ direction, i.e. left-right spatial mesh sweeps. This method could be generalised, for example by using a weighted upwind method instead of an average upwind method. For example, one could use $\psi_{IN}^i = \theta\psi_2^{i-1} + (1-\theta)\,\psi_1^i$ as the incoming value on the boundary. This weighted upwind method is a θ-method, and has the advantage that the scheme can be switched back to standard DG (by setting $\theta = 0$) without having a separate implementation for this. Also, it would be interesting to study the effect of different values of θ (different degrees of upwinding). However, the weighted average method is left as a future topic and not considered in this Thesis. The matrix equation modified for the upwind average DG scheme may be written, for the $\mu > 0$ direction for mesh element i as:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\psi_2^{i-1} + \psi_1^i) \\ -\psi_2^i \end{bmatrix}. \tag{2.59}$$

Note that ψ_1^i and ψ_2^i that appear on the right hand side are part of the element i solution. Taking them over to the left hand side of this equation, to absorb them

into the solution to the matrix equation, one will obtain:

$$\begin{pmatrix} a_{11} - \frac{1}{2} & a_{12} \\ a_{21} & a_{22} + 1 \end{pmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\psi_2^{i-1} \\ 0 \end{bmatrix}, \tag{2.60}$$

in which ψ_2^{i-1} is the flux flowing out of the previous mesh cell on the $\mu > 0$ sweep. Hence it is convenient to define the matrix:

$$L = \begin{pmatrix} -\frac{1}{2} & 0 \\ 0 & 1 \end{pmatrix}. \tag{2.61}$$

For the $\mu < 0$ directions, the matrix equation for the upwind average DG scheme becomes:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} = \begin{bmatrix} -\psi_1^i \\ \frac{1}{2}(\psi_2^i + \psi_1^{i+1}) \end{bmatrix}. \tag{2.62}$$

As ψ_1^i and ψ_2^i are part of the element i solution, they are taken over to the left hand side of the equation in order for them to be absorbed into the solution. The equation, with this modification becomes:

$$\begin{pmatrix} a_{11} + 1 & a_{12} \\ a_{21} & a_{22} - \frac{1}{2} \end{pmatrix} \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2}\psi_1^{i+1} \end{bmatrix}, \tag{2.63}$$

in which ψ_1^{i+1} is the flux flowing out of the previous mesh cell on the $\mu < 0$ sweep. Hence it is convenient to define the matrix:

$$L = \begin{pmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{pmatrix}. \tag{2.64}$$

Using these definitions of L, which switch from one to the other according to the spatial mesh sweep, one may write down a general matrix equation describing the upwind average DG method:

$$\left[\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} + L \right] \begin{bmatrix} \psi_1^i \\ \psi_2^i \end{bmatrix} = \mathbf{b_i}, \tag{2.65}$$

where $\mathbf{b_i}$ is the vector containing the right hand side of the equation for element i. The definition of this also varies according to spatial mesh sweep. For mesh

element i, the vector $\mathbf{b_i}$ may be defined as:

$$\mathbf{b_i} = \begin{cases} (\frac{1}{2}\psi_2^{i-1}, 0)^T, & \mu > 0; \\ (0, \frac{1}{2}\psi_1^{i+1})^T, & \mu < 0. \end{cases} \tag{2.66}$$

In this sweep-dependent definition of $\mathbf{b_i}$, the angular flux on the outlet node of the previous element on the sweep is used. Any other sources that are present can simply be added on the the vector to complete the right hand side of the equations.

2.8.4. Petrov-Galerkin Stabilisation

The methods implemented so far achieve a basic solution that is very accurate, simple to implement and practical computationally. However, in demanding radiation transport computations robustness of the solution scheme is also very important, and hence there is need for some form of stabilisation in the standard DG method which is prone to oscillate. This section describes a Petrov-Galerkin (PG) stabilisation scheme for the methods implemented so far and discusses its implementation. Stabilisation is achieved in the standard DG method by adding to the matrix equation an upwind term that contains artificial dissipation:

$$\int_{V_E} dV \left(\mathbf{N}_i \mathscr{R} - \mathbf{N}_i \mu \frac{\partial \mathbf{p} \mathscr{R}}{\partial x} \right) = 0, \tag{2.67}$$

where $\mathbf{p}$ is a stabilisation parameter. Choices of the value for $\mathbf{p}$ are discussed in section (3), where expressions are derived regarding optimisation of the length across which dissipation is introduced to improve stability. The length-scale itself may be regarded as a parameter that scales the stabilisation, in the extreme being equal to Δx where Δx is the width of an element. Excessive stabilisation may reduce accuracy of solution, and $\mathbf{p}$ may be used to scale out the Petrov-Galerkin term where it is not necessary, while recovering it where it is needed. We next apply Green's theorem to the volume integral containing the derivative in equation (2.67), as done previously. This brings in a contribution from the element surface Γ_E that contains the residual $\mathscr{R}$ in the surface integral:

$$\int_{V_E} dV \mathbf{N}_i(x) \mathscr{R} + \mu \int_{V_E} dV \mathbf{N}_{x_i}(x) \mathbf{p} \mathscr{R} - \int_{\Gamma_E} d\Gamma n_x \cdot \mu \mathbf{N}_i(x) \mathbf{p} \mathscr{R} = 0. \tag{2.68}$$

Arising at the boundaries between adjacent nodes, this surface term provides additional coupling between elements. However, it contains a derivative inside the residual. This is problematic as it is on the element surface where continuity is not enforced. A method for including it in the discretisation is considered in section (2.8.5). Alternatively, one may set the surface term that contains the residual, to be zero. This is a natural boundary condition of the equation but only includes the added dissipation internal to the element (where the finite element space is continuous). This method is know in the present work as the Within-Element Petrov-Galerkin (WEPG) method and is described in section (2.8.7). We first consider a method that discretises the surface integral containing the derivative. This is known as the Between-Element Petrov-Galerkin (BEPG) method.

2.8.5. Between Element PG Approach

An alternative to setting the surface term in equation (2.68) to be zero, is to discretise it along the spatial discontinuity. This section formulates a method that accomplishes this, the idea being that the added Petrov-Galerkin dissipation is included in coupling neighbouring between elements. The second order term with which we are concerned, that is the term that is added to the discretisation to achieve PG stabilisation, is:

$$\frac{\partial(\mathbf{p}\mathscr{R})}{\partial x} = \frac{\partial}{\partial x}\left(\mathbf{p}\frac{\partial \psi}{\partial \mathbf{x}}\right), \tag{2.69}$$

where we have substituted the expansion of $\mathscr{R}$ ignoring all terms except the advection term, as these are easily added later. We begin with the standard DG projection across a single element E in which we multiply the expression by one of the discontinuous basis functions $\mathbf{N}_i$ and integrate across the element domain. Applying Green's theorem to equation (2.69) the surface terms are produced and we obtain:

$$\int_{V_E} dV \mathbf{N}_i \frac{\partial \mathbf{p} \partial \psi}{\partial x^2} = -\int_{V_E} dV \mathbf{N}_{i_x} \mathbf{p} \frac{\partial \psi}{\partial x} + \oint_{\Gamma_E} d\Gamma \mathbf{N}_i n_x \cdot \mathbf{p} \frac{\partial \psi}{\partial x}. \tag{2.70}$$

In these integrals, V_E defines the volume of the element E domain and n_x is a unit vector normal to the element E boundary, in the x-direction. Note that n_x will differ in sign at the inbound and outbound faces, generally assuming negative sign at the inlet as the normal here will be anti-parallel to the direction of particles entering the element. In order to take into account the inter-element coupling, we define

a jump condition. We consider element F that is adjacent to element E, and com-

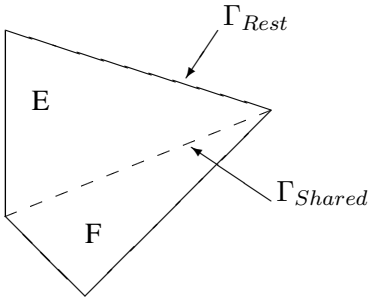


Figure 2.5.: The Between Element Petrov-Galerkin method.

bine the domain of element F with the domain of element E to form a new domain of volume V_{EF} illustrated in figure (2.5). In this figure, $\Gamma_{E\cap F}$ is the contribution from the surface that element E shares with adjacent element F as illustrated. $\Gamma_{\partial E}$ is the remaining contribution from the element E surface, which excludes the surface of intersection with element F. We note that in equation (2.70), the surface part is problematic because it requires evaluation of a derivative at a point in space where it is discontinuous. The objective is to evaluate this derivative $\frac{\partial \psi}{\partial x}$ on the surface $\Gamma_{E\cap F}$, along the discontinuous boundary shared by element E and element F in the centre of the combined domain. This is illustrated in figure (2.6). Continuous finite element basis functions $\mathbf{M}_j$ span the combined domain encapsu-

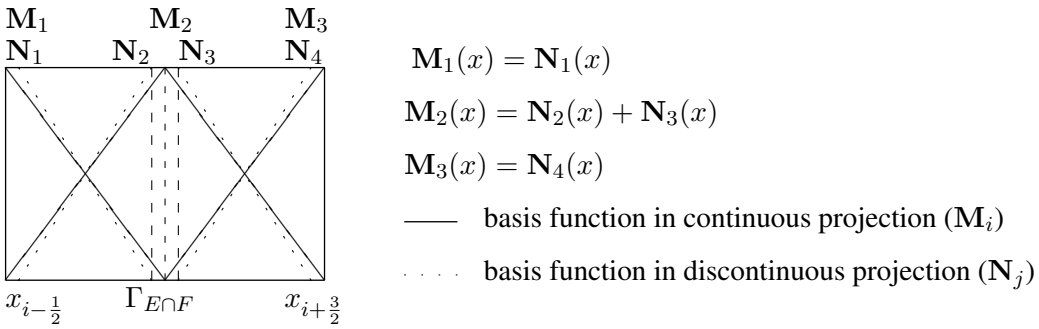


Figure 2.6.: Evaluating the boundary derivative in the Between Element Petrov-Galerkin method.

lating the surface $\Gamma_{E\cap F}$. These basis functions may be regarded as the continuous counterparts to the basis functions $\mathbf{N}_i$ used in the discontinuous projection across a single element, such as E or F. We approximate the derivative across the combined domain with an unknown continuous parameter q_x. The discretisation of q_x is a 3

component column vector q_j, and q_x has the continuous finite element expansion:

$$q_x = \sum_j \mathbf{M}_j q_j \approx \frac{\partial \psi}{\partial x}. \tag{2.71}$$

Applying a Galerkin projection of q_x across the continuous space using continuous finite elements that span the intersection we obtain:

$$\int_{V_{EF}} dV \mathbf{M}_j q_x \approx \int_{V_{EF}} dV \mathbf{M}_j \frac{\partial \psi}{\partial x} = -\int_{V_{EF}} dV \frac{\partial \mathbf{M}_j}{\partial x} \psi + \oint_{\Gamma_{EF}} d\Gamma \mathbf{M}_j n_x . \psi = B_{x_{EF}} \psi_{EF}, \tag{2.72}$$

where V_{EF} denotes the volume of the combined domain made up of elements E and F and Γ_{EF} denotes the whole surface of the combined domain. We have introduced the matrix $B_{x_{EF}}$, element ij of which is defined to be:

$$B_{x_{EF_{ij}}} = -\int_{V_{EF}} dV \mathbf{M}_{i_x} \mathbf{N}_j + \oint_{\Gamma_{EF}} d\Gamma_{EF} \mathbf{M}_i n_x \cdot \mathbf{N}_j, \tag{2.73}$$

to allow the surface integration to be performed only along the shared boundary $\Gamma_{E \cap F}$. We have also introduced column vector ψ_{EF} defined to be:

$$\psi_{EF} = (\psi_1 \; \psi_2 \; \psi_3 \; \psi_4)^T, \tag{2.74}$$

where ψ_1 and ψ_2 form the solution vector for element E and ψ_3 and ψ_4 form the downwind solution vector associated with element F. Using equation (2.71), a Galerkin projection of q_x across the combined volume using continuous finite elements may also be obtained via:

$$\int_{V_{EF}} dV \mathbf{M}_j q_x = \int_{V_{EF}} dV \mathbf{M}_j \sum_k \mathbf{M}_k q_k = D_{EF} \mathbf{q_{EF}}, \tag{2.75}$$

where D_{EF} is the local mass matrix for the combined domain, element ij of which is defined to be:

$$D_{EF_{ij}} = \int_{V_{EF}} dV \mathbf{M}_i \mathbf{M}_j, \tag{2.76}$$

and the column vector $\mathbf{q_{EF}}$ defined to be:

$$\mathbf{q_{EF}} = (q_1\ q_2\ q_3)^T. \tag{2.77}$$

Combining equation (2.72) with equation (2.75) we obtain the following matrix equation for $\mathbf{q}$ that applies across the combined domain:

$$\mathbf{q} = D_{EF}^{-1} B_{x_{EF}} \psi_{EF}. \tag{2.78}$$

We note that, for basis functions of the form (2.35) the elements of the mass matrix assume the values:

$$D = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & \frac{1}{6} & \frac{1}{3} \end{pmatrix}, \tag{2.79}$$

in isoparametric coordinates. These must be scaled via the Jacobian in order to perform the numerical integrations in local coordinates. The values quoted apply in the domain of the quadrature. Typically, the Jacobian will be $\frac{\Delta x}{2}$ where Δx is the width of either element E or the width of element F depending on which part of the mass matrix is under evaluation. In order to solve for column vector $\mathbf{q}$, we must first evaluate $B_{x_{EF}}$ in the local coordinates. Since we are not dealing with non-conforming elements and discontinuous basis functions, the continuous basis functions $\mathbf{M}_i$ can be formed from a linear combination of their discontinuous counterparts $\mathbf{N}_j$. Hence we can write $\mathbf{M}_i$ = $\mathbf{N}_p$ + $\mathbf{N}_q$ where subscripts p and q relate to subscript i according to table (2.1). Index i refers to a continuous

i	p	q
1	1	-
2	2	3
3	-	4

Table 2.1.: Continuous and discontinuous node numbers in the Between Element PG method.

basis function, and indices p and q each refer to a discontinuous basis function. It is clear that $\mathbf{N}_p$ is non-zero only in the domain of element E, while $\mathbf{N}_q$ is non-zero only in the domain of element F. This enables us to derive an expression for matrix $B_{x_{EF}}$ such that the surface integration is performed only along the shared

boundary $\Gamma_{E\cap F}$ purely in terms of the discontinuous basis functions $\mathbf{N}_p$ and $\mathbf{N}_q$:

$$\int_{V_{EF}} dV\mathbf{M}_i\frac{\partial\psi}{\partial x} = \int_{V_{EF}} dV\mathbf{N}_p\frac{\partial\psi}{\partial x} + \int_{V_{EF}} dV\mathbf{N}_q\frac{\partial\psi}{\partial x} = -\int_{V_E} dV\mathbf{N}_{p_x}\psi + \oint_{\Gamma_{EF}} d\Gamma\mathbf{N}_p n_x{\cdot}\psi - \int_{V_F} dV\mathbf{N}_{q_x}\psi + \oint_{\Gamma_{EF}} d\Gamma\mathbf{N}_q n_x{\cdot}\psi, \quad (2.80)$$

where we have used the fact that discontinuous basis function $\mathbf{N}_p$ is non-zero only in element E and discontinuous basis function $\mathbf{N}_q$ is non-zero only in element F. Considering the element E contribution to the combined domain in the continuous projection, in isolation from the element F contribution, we have:

$$\int_{V_E} dV\mathbf{N}_p\frac{\partial\psi}{\partial x} = -\int_{V_E} dV\mathbf{N}_{p_x}\psi + \oint_{\Gamma_{E\cap F}} d\Gamma\mathbf{N}_p n_{x_E}\psi_{_F} + \oint_{\Gamma_{EF}} d\Gamma\mathbf{N}_p n_x\psi. \quad (2.81)$$

In the above, $\psi_{_F}$ is the nodal value of the flux on the element F side of the boundary $\Gamma_{E\cap F}$ that element E shares with element F. n_{x_E} is the outward unit vector normal to element E at the boundary $\Gamma_{E\cap F}$. The choice regarding which side of the $\Gamma_{E\cap F}$ boundary on which to evaluate ψ in the $\Gamma_{E\cap F}$ surface integral inserts the jump condition which provides the coupling between element E and element F. By taking $\psi_{_F}$ we couple element E to the downwind solution at the element E surface. The Γ_{EF} surface integral is zero when p=2, due to the $\mathbf{N}_2$ basis function being zero at this point. In this case, the only contribution to the element surface is that of the surface $\Gamma_{E\cap F}$ at which $\mathbf{N}_2$ evaluates to unity. In contrast, the integral on the surface $\Gamma_{E\cap F}$ becomes zero when p=1 for which case the Γ_{EF} integral becomes the only non-zero contribution along the boundary. A similar expression to 2.81 can be obtained for element F, in which case p is substituted for q and the unit normal assumes opposite sign. We write for element F:

$$\int_{V_F} dV\mathbf{N}_q\frac{\partial\psi}{\partial x} = -\int_{V_F} dV\mathbf{N}_{q_x}\psi + \oint_{\Gamma_{E\cap F}} d\Gamma\mathbf{N}_q n_{x_F}\psi_{_E} + \oint_{\Gamma_{EF}} d\Gamma\mathbf{N}_q n_x\psi. \quad (2.82)$$

Rearranging equations (2.81) and (2.82) in terms of the Γ_{EF} surface integrals, we may substitute them into equation (2.80) eliminating the surface of the combined domain. This leaves only the integrals that lie along the shared boundary between element E and element F. Hence:

$$\int_{V_{EF}} dV N_i \frac{\partial \psi}{\partial x} = \int_{V_E} dV \mathbf{N}_p \frac{\partial \psi}{\partial x} - \oint_{\Gamma_{E\cap F}} d\Gamma \mathbf{N}_p n_{x_E} \psi_F + \int_{V_F} dV \mathbf{N}_q \frac{\partial \psi}{\partial x} - \oint_{\Gamma_{E\cap F}} d\Gamma \mathbf{N}_q n_{x_F} \psi_E, \tag{2.83}$$

eliminating the Γ_{EF} surface integrals around the boundary of the combined domain. We note from equation (2.72) that the ith row of equation (2.83) is the ith row of matrix $B_{x_{EF}}$. Hence we recast equation (2.83) in matrix notation in terms of $B_{x_{EF}}$ generating the system of equations:

$$B_{x_{EF}} = \begin{pmatrix} a_{1,1} & a_{1,2} & 0 & 0 \\ a_{2,1} & a_{2,2}+1 & a_{3,3}-1 & a_{3,4} \\ 0 & 0 & a_{4,3} & a_{4,4} \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{pmatrix}, \tag{2.84}$$

in which the non-zero element $a_{i,j}$ is defined by the integral:

$$a_{i,j} = \int_{V_{EF}} dV \mathbf{N}_i(x) \frac{\partial \mathbf{N}_j(x)}{\partial x}. \tag{2.85}$$

In these equations, ψ_1 and ψ_2 are the fluxes at nodes 1 and 2 of the element being solved for (i.e. element E), and ψ_3 and ψ_4 are the fluxes at nodes 1 and 2 of the element downwind of the element being solved for (i.e. element F). To discretise the derivative at the inlet node of element E (i.e. q_1), row 2 of matrix $B_{x_{DE}}$ is evaluated; in order to discretise the derivative at the element E outlet (i.e. q_2) row 2 of matrix $B_{x_{EF}}$ is evaluated. The notation used for the discontinuous basis functions $\mathbf{N}_i$ and their derivatives $\mathbf{N}_{i_x}$ is that used in figure (2.6). This allows us to determine $\mathbf{q}$ by inverting the mass matrix (2.79). We note that, for basis functions of the form (2.35) the elements of the B_x matrix assume the values:

$$B_x = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \\ 0 & 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, \tag{2.86}$$

in isoparametric coordinates. The Jacobian is introduced to the numerical integrals to allow the matrix elements to be evaluated in local coordinates. The downwind part of the flux vector this matrix operates on must be taken from the previous iteration of the solver. The surface integral in equation (2.70) may be written:

$$\oint_{\Gamma_E} d\Gamma \mathbf{N}_i n_x . \frac{\partial \psi}{\partial \psi} = \oint_{\Gamma_E} d\Gamma \mathbf{N}_i n_x . p\mathbf{q} = \sum_j \oint_{\Gamma_E} d\Gamma \mathbf{N}_i n_x . p\mathbf{M}_j \mathbf{q_j}, \tag{2.87}$$

and by forming the matrix for each element F adjacent to E:

$$Q_{EF_{ij}} = \oint_{\Gamma_{E \cap F}} d\Gamma \mathbf{N}_i n_x . p\mathbf{M}_k, \tag{2.88}$$

we obtain the discretisation of the surface integral in equation (2.70):

$$Q_{EF} D_{EF}^{-1} B_{x_{EF}} \psi. \tag{2.89}$$

2.8.6. Symmetric Discretisation of Second Order Terms

The second order operator may be written:

$$\nabla \cdot \nabla \psi = \nabla \cdot \mathbf{q_x}, \tag{2.90}$$

where

$$\mathbf{q_x} = \sum_j N_j q_j. \tag{2.91}$$

Considering the Galerkin projection of equation (2.90) across element E using continuous finite elements we have:

$$\begin{aligned} \int_E dV N_i \nabla \cdot \mathbf{q_x} &= -\int_V dV \nabla N_i \cdot \sum_j N_j q_j + \int_\Gamma d\Gamma N_i \cdot \mathbf{n_x} \mathbf{q_x} \\ = -\int_V dV \nabla N_i \cdot \sum_j N_j q_j &+ \int_{\Gamma_{IN}} d\Gamma N_i \cdot \mathbf{n_x} \mathbf{q_x} + \int_{\Gamma_{OUT}} d\Gamma N_i \cdot \mathbf{n_x} \mathbf{q_x} \\ &= B_x^T \underline{\mathbf{q}}, \end{aligned} \tag{2.92}$$

where the surface integral has been split into separate inbound and outbound contributions, as previously. Recasting this equation in matrix notation we obtain:

$$\int_E dV N_i \nabla \cdot \mathbf{q_x} = B_x^T \underline{\mathbf{q}} = \underline{b}, \tag{2.93}$$

where $\underline{b}$ is the source vector $(b_1\ b_2)^T$, $\underline{\mathbf{q}} = (q_1\ q_2)^T$ and the elements of the B_x^T matrix assume the values:

$$B_x^T = \begin{pmatrix} a_{1,1}+1 & a_{1,2} \\ a_{2,1} & a_{2,2}-1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{3}{2} \end{pmatrix}, \tag{2.94}$$

in which element $a_{i,j}$ is defined as:

$$a_{i,j} = \int_{V_E} dV - \mathbf{N}_i(x) \frac{\partial \mathbf{N}_j(x)}{\partial x}, \tag{2.95}$$

in local coordinates. We now consider the Galerkin projection of $\mathbf{q_x}$ across a single element:

$$\int_E dV N_i \mathbf{q_x} = \int_V dV N_i \sum_j N_j q_j = Q_E \underline{\mathbf{q}}, \tag{2.96}$$

where $\underline{\mathbf{q}} = (q_1\ q_2)$ and Q_E is the local mass matrix. In local coordinates the matrix Q_E may be written:

$$Q_E = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{pmatrix}, \tag{2.97}$$

in which element $a_{i,j}$is obtained from:

$$a_{i,j} = \int_{V_E} dV \mathbf{N}_i(x) \mathbf{N}_j(x). \tag{2.98}$$

We note from equation (2.90) that

$$\int_E dV N_i \mathbf{q_x} = \int_E dV N_i \nabla \psi = B_x \underline{\mathbf{\Psi}}, \tag{2.99}$$

and so equations (2.96) and (2.99) combine to give:

$$Q_E \underline{\mathbf{q}} = B_x \underline{\mathbf{\Psi}} \longrightarrow \underline{\mathbf{q}} = Q_E^{-1} B_x \underline{\mathbf{\Psi}}. \tag{2.100}$$

Substituting equation (2.100) into equation (2.93) $\underline{\mathbf{q}}$ is eliminated:

$$B_x^T Q_E^{-1} B_x \underline{\mathbf{\Psi}} = \underline{\mathbf{b}}, \tag{2.101}$$

where B_x is evaluated simply by forming the transpose of B_x^T:

$$B_x = \left[B_x^T\right]^T = \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{3}{2} \end{pmatrix}. \tag{2.102}$$

2.8.7. Within Element PG Approach

The first stage in implementing the full PG stabilisation scheme was to set the surface term in equation (2.68) to be zero. This obtains the internally stabilised form of the standard DG representation, in absence of any inter-element coupling:

$$\int_{V_E} dV \mathbf{N}_i \mathscr{R} + \int_{V_E} dV \mathbf{N}_{i_x} (p\mathscr{R}) = 0. \tag{2.103}$$

The second integral in equation (2.103), which provides the stabilisation, is internal to the element domain and we may therefore substitute trial functions (2.34) into the expansion of $\mathscr{R}$ inside the second integral. The first term, however must be integrated by parts and treated as previously, to produce the remaining surface contribution. Considering only the stabilisation term, we insert $\mathscr{R}$ and the trial functions to obtain:

$$\begin{aligned} \int_{V_E} dV \mathbf{N}_{i_x} (pR) &= \int_{V_E} dV \mathbf{N}_{i_x} p \left(\frac{\partial \psi}{\partial x} + \sigma_i \psi \right) \\ &= \int_V dV \mathbf{N}_{i_x} p \left(\sum_j \mathbf{N}_j \psi_j + \sigma_i \sum_j \mathbf{N}_j \psi_j \right). \end{aligned} \tag{2.104}$$

Performing the numerical integration, we obtain from equation (2.104) the PG stabilisation matrix. We add this on to the unstabilised standard DG matrix equation

to obtain:

$$\left[\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} + \mathbf{S} + \mathbf{L}\right] \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} = \mathbf{b}, \tag{2.105}$$

in which element $a_{i,j}$ of the above matrix is defined to be:

$$a_{i,j} = \int_{V_E} dV \left(-\frac{\partial \mathbf{N}_i(x)}{\partial x} \mathbf{N}_j(x) + \sigma(x) \mathbf{N}_i(x) \mathbf{N}_j(x) \right), \tag{2.106}$$

and in which the matrix $\mathbf{S}$ provides within element PG stabilisation and is defined to be:

$$\mathbf{S} = \mathbf{p} \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}. \tag{2.107}$$

The element $a_{i,j}$ of the above stabilisation matrix is defined to be:

$$a_{i,j} = \int_{V_E} dV \left(-\left(\frac{\partial \mathbf{N}_i(x)}{\partial x}\right) \left(\frac{\partial \mathbf{N}_j(x)}{\partial x}\right) + \sigma(x) \frac{\partial \mathbf{N}_i(x)}{\partial x} \mathbf{N}_j(x) \right). \tag{2.108}$$

Matrices $\mathbf{L}$ and $\mathbf{b}$ are defined as previously, according to the sign of μ. Of course, their definitions also vary depending on whether the upwinding is full or averaged as described in sections (2.8.1) and (2.8.3).

2.8.8. Quadratic Bubble Approach

Sub-grid scale bubble functions are polynomial basis functions that are defined on the interiors of a finite element. They vanish on the boundaries of the element and add a multi-scale aspect to the discretisation that has the effect of stabilising the computational solution. This has lead to their use in oscillation control and relates them to PG methods, with which they have similar properties. They also enhance the element by improving resolution close to the boundary. In many flow problems, boundary layers occur near an interface challenging the standard methods. In problems such as these, sub-grid enrichment of the solution is often required. One implements these so called bubble functions by adding a third node (in the case of low order one-dimensional elements) that has the same basis function associated with it as the centre-node basis function that one uses in a standard quadratic element. Enriching the finite element space in this way therefore gives rise to a three node element, in which the two existing shape functions are those of a standard linear element. Extension of this approach to higher dimen-

sions has already been demonstrated to be straightforward (Candy, 2010). The bubble functions have zero value on the element boundary; however they give the element much improved properties in the layers close to the edge of the element. One includes the bubble functions by extending the summation over the finite element bases to include the extra polynomial, enhancing the underlying scheme. It has been shown (Brezzi et al., 1998) that including even low order bubble functions in the finite element space results in stabilisation in convective and diffusive driven problems. Sub-grid scale bubble functions are potentially useful within a DG framework, as the bubble can easily be up-winded away from the centre of the element. The approximate solution to the problem is expressed as the sum of a discontinuous function $\psi_D(x)$ and a continuous function $\psi_C(x)$:

$$\psi(x) = \psi_D(x) + \psi_C(x). \tag{2.109}$$

Expanding equation (2.109) in a finite element representation for a single element that includes the quadratic bubble function one has:

$$\psi(x) = \mathbf{N}_1(x)\phi_1 + \mathbf{N}_2(x)\phi_2 + \mathbf{N}_3(x)\phi_3, \tag{2.110}$$

in which the vector $\Phi = (\phi_1, \phi_2, \phi_3)^T$ contains the solution for the three nodes the element contains. The finite element shape functions may be defined, in parametric coordinates, to be:

$$\mathbf{N}_1(x) = \frac{1-x}{2}, \tag{2.111}$$

$$\mathbf{N}_2(x) = 1 - x^2, \tag{2.112}$$

$$\mathbf{N}_3(x) = \frac{x+1}{2}. \tag{2.113}$$

Shape functions $\mathbf{N}_1(x)$ and $\mathbf{N}_3(x)$ are the same as those used for a one-dimensional linear element type. The shape function $\mathbf{N}_2(x)$ is the added bubble function. It is convenient to express these functions parametrically so that an integration rule can be used to assemble the matrices. A Galerkin weighting may then be used as previously, and a 3×3 stencil obtained for each element. The continuous shape function $\mathbf{N}_2(x)$ that defines the bubble function may be modified, using the material cross-section, to upwind the bubble.

2.9. Taylor Analysis

In this section, Taylor analysis is used to explore the convergence properties of some of the one-dimensional methods. This involves expanding the discrete one-dimensional transport equation, for each method and comparing the expansion to that of an analytical result. The number of terms to which the two series agree indicates the truncation order of the method. The schemes that have been subjected to this analysis are the standard discontinuous Galerkin method, the upwind average method and the within-element Petrov-Galerkin method. The diamond difference method has also been included in the analysis for benchmarking, as it remains a popular scheme.

2.9.1. Diamond Difference

The diamond difference equation may be written as:

$$\frac{\phi_{OUT}}{\phi_{IN}} = \frac{1-h}{1+h}, \tag{2.114}$$

where for an element of width Δx, the length scale $h = \frac{\sigma \Delta x}{2|\mu|}$ is the number of mean free paths across the element in direction μ. ϕ_{OUT} is the flux leaving the the outlet node of the element and ϕ_{IN} is the flux entering on the inbound node. Across this element, the transport equation may be written:

$$|\mu| \frac{\partial \psi(x)}{\partial x} + \sigma(x)\phi(x) = 0, \tag{2.115}$$

and the exact solution may be written:

$$\phi_{OUT} = \phi_{IN} e^{-2h} \longrightarrow \frac{\phi_{OUT}}{\phi_{IN}} = e^{-2h}, \tag{2.116}$$

and hence the approximation made when one applies the diamond difference scheme is:

$$\frac{1-h}{1+h} \approx e^{-2h}. \tag{2.117}$$

One is now in a position to obtain the spatial truncation order of this method, by expanding both $\frac{1-h}{1+h}$ and e^{-2h} in a Taylor series. The order to which the two

expansions match defines the truncation order. e^{-2h} is given by the infinite series:

$$\begin{aligned} e^{-2h} &= 1 - 2h + \frac{4h^2}{2!} - \frac{8h^3}{3!} + \frac{16h^4}{4!} - \frac{32h^5}{5!} + ... \\ &= 1 - 2h + 2h^2 - \frac{4h^3}{3} + \frac{2h^4}{3} - \frac{4h^5}{15} + ... \end{aligned} \tag{2.118}$$

Expanding $\frac{1-h}{1+h}$ in an infinite series:

$$\begin{aligned} \frac{1-h}{1+h} &= (1-h)(1+h)^{-1} = (1-h)(1-h+h^2-h^3+h^4-h^5+...) \\ &= 1 - 2h + 2h^2 - 2h^3 + 2h^4 - 2h^5 + ... \\ &= \mathcal{O}(2)e^{-2h}. \end{aligned} \tag{2.119}$$

Thus the diamond difference method is at most, second order in terms of the spatial truncation.

2.9.2. Linear Discontinuous

The linear discontinuous equation may be arranged into the form:

$$\frac{\phi_2}{\phi_{IN}} = \frac{3-2h}{3+4h+2h^2}. \tag{2.120}$$

Note that the weighting of the equation may alter this expression, for example if a different test function is used to formulate the method. The length scale has the same definition as for the diamond difference case, that is $h = \frac{\sigma \Delta x}{2|\mu|}$ and Δx the width of the element. As done previously, the right hand side of the equation (2.120) may, after some manipulation, easily be expanded in an infinite series:

$$\begin{aligned} \frac{3-2h}{3+4h+2h^2} &= (3-2h)(\frac{1}{3} - \frac{4h}{9} + \frac{10h^2}{27} - \frac{16h^3}{81} + ...) \\ &= 1 - 2h + 2h^2 - \frac{4h^3}{3} + ... \\ &= \mathcal{O}(3)e^{-2h}. \end{aligned} \tag{2.121}$$

Therefore, the linear discontinuous finite element method is third order in terms of the spatial truncation error. There are efficiency arguments in favour of improved truncation error, as fewer elements are needed for given error in the solution.

2.9.3. Petrov-Galerkin

The Petrov-Galerkin method may be written as:

$$\frac{\phi_2}{\phi_{IN}} = \frac{6-5h}{6+7h+2h^2}. \tag{2.122}$$

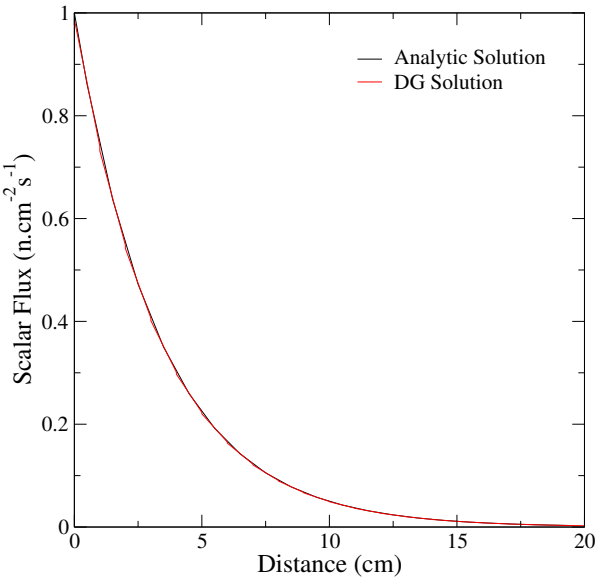
The left hand side of this equation is expanded in an infinite series as done previously for the other methods, leading to:

$$\begin{aligned} \frac{6-5h}{6}\left(1+\frac{7h+2h^2}{6}\right)^{-1} &= \frac{6-5h}{6}\left(1-2h+2h^2-\frac{5h^3}{3}+...\right) \\ &= \mathcal{O}(2)e^{-2h}. \end{aligned} \tag{2.123}$$

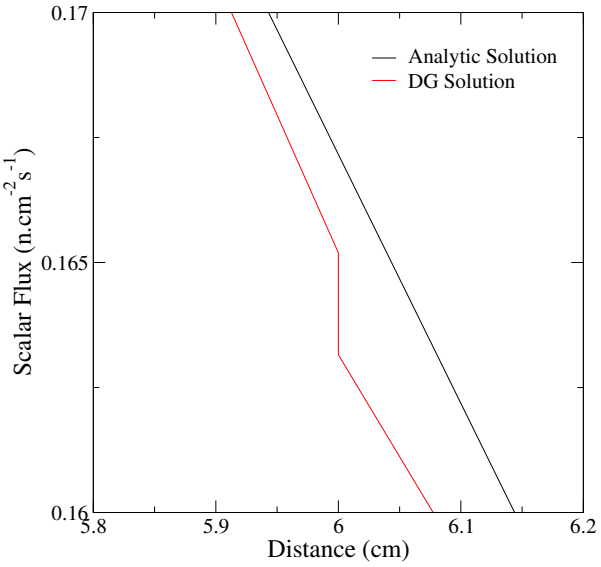
Including Petrov-Galerkin weighting in the discretisation therefore reduces the truncation order of the linear discontinuous finite element method, to second order.

2.10. Numerical Examples

In this section results from some of the one-dimensional schemes that have been discussed in the present chapter are presented. The standard DG method has been compared to an analytical solution for a pure absorber in one-dimension. The basic scheme provides an acceptable representation in this type of problem, as shown in figure (2.7). This problem used 20 elements on a 20.0cm mesh, and shows the computational solution at each node where the exact solution has also been calculated. The upwind average discontinuous method, which is similar to DG but uses an average value in the surface terms, is shown in figure (2.8). This method is apparently not quite as accurate as the standard form and it exhibits a larger discontinuity than standard DG. It also lacks any form of stabilisation in the discretisation, which makes it less robust for more demanding problems. The Within-Element Petrov- Galerkin method adds upwind stabilisation into the discretisation, and exhibits a similar numerical accuracy to the standard DG scheme in this problem. The Within-Element Petrov-Galerkin method is presented in figure (2.9) and compared to the other one-dimensional schemes. It is suggested for future work, that the numerical convergence rate of each scheme is obtained and compared to that suggested by the Taylor analysis.

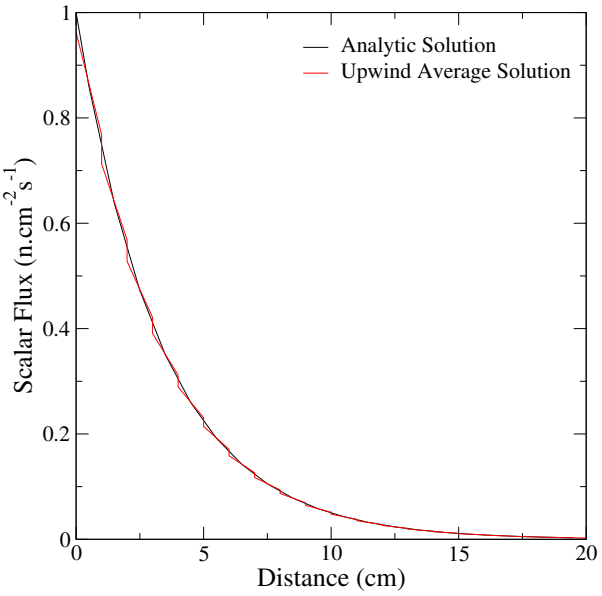


(a) Standard DG versus analytic solution

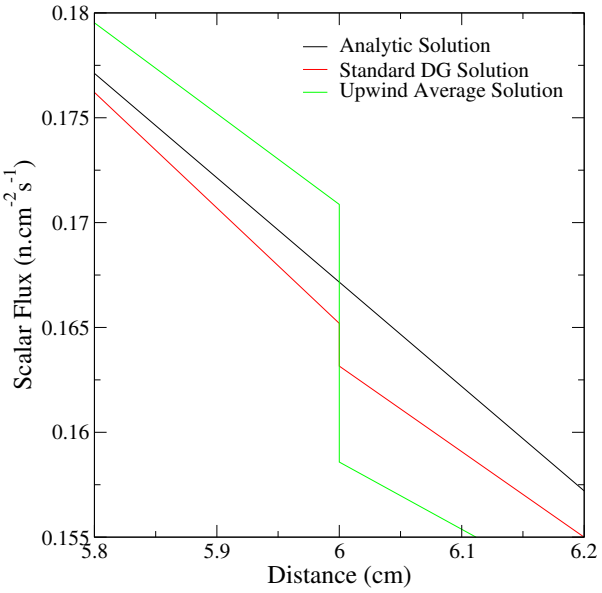


(b) Nodal discontinuities in standard DG solution

Figure 2.7.: Standard discontinuous Galerkin solution in a one-dimensional absorber problem using a 20.0 cm mesh and 20 elements. The analytic result is well represented.

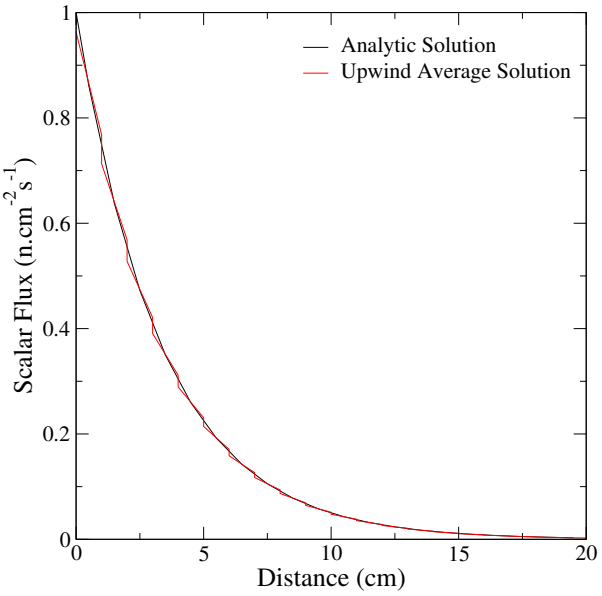


(a) Upwind Average DG versus analytic solution

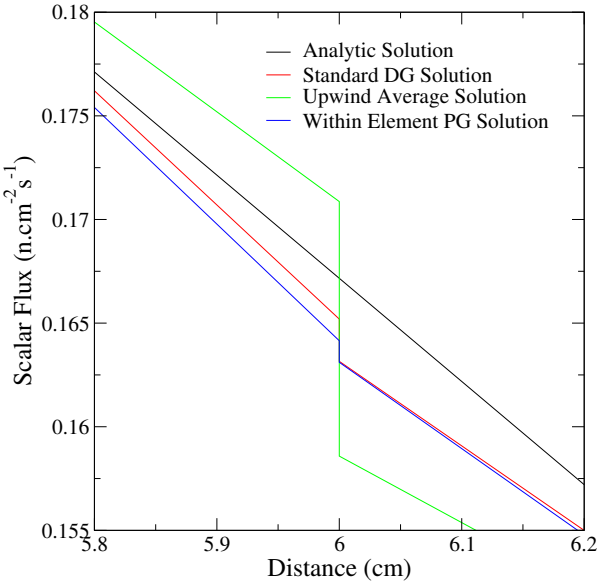


(b) Nodal discontinuities in Upwind Average DG solution

Figure 2.8.: Upwind Average discontinuous Galerkin solution in a one-dimensional absorber problem using a 20.0 cm mesh and 20 elements. The analytic result is well represented, but perhaps less accurate than DG although the discontinuities appear smaller.



(a) Within-Element Petrov-Galerkin versus analytic solution



(b) Nodal discontinuities in Within-Element Petrov-Galerkin solution

Figure 2.9.: Within-Element Petrov-Galerkin solution in a one-dimensional absorber problem using a 20.0 cm mesh and 20 elements. The analytic result is well represented.

2.11. Conclusions

In this chapter, the Boltzmann Transport Equation (BTE) was introduced, and consideration given to several methods of performing the angular discretisation. The angular discretisation schemes were followed by a description of the spatial discretisation, for which discontinuous finite elements are used with Galerkin weighting. In this Thesis, the spatial discretisation is done on an orthogonal mesh in Cartesian geometry. A Riemann approach to implementing the surface integrals in the discrete equations was introduced. This method was selected because it is very general allowing different basis functions in direction of particle travel to be used with the spatial discretisation scheme. Some discontinuous methods, including the standard discontinuous Galerkin method, were discussed as background to the present work. The additional schemes that were discussed included an upwind average DG scheme, that differs from the standard discontinuous Galerkin scheme by averaging the incoming information on the boundary of each element; two Petrov-Galerkin methods that differ from one another in how the surface terms are treated; and a scheme that adds a sub-grid bubble function to the standard discontinuous Galerkin discretisation. One of the Petrov-Galerkin schemes adds dissipation internal to an element, and is referred to in the present work as Within-Element Petrov-Galerkin (WEPG). The second Petrov Galerkin scheme adds dissipation both internal to an element and also into the coupling between elements. This second scheme is referred to as Between-Element Petrov-Galerkin (BEPG). The WEPG formulation was selected to become the topic of further study and further development in the chapter that follows. The WEPG scheme was chosen because it is a compromise between complexity of implementation, and accuracy; it improves on the standard DG by adding artificial dissipation to the discretisation. The use of this for oscillation control is to be investigated in the chapters that follow. The WEPG scheme is also far easier to implement than more sophisticated PG schemes, such as the BEPG scheme.

CHAPTER 3

LINEAR OPTIMAL METHODS

Synopsis

The standard discontinuous Galerkin method performs exceptionally well in a very broad range of radiation transport problems. C^0 continuity of the solution is not enforced, allowing the capture of shocks and sharp gradients in the problem. However, there are cases in which standard discontinuous Galerkin schemes will oscillate and behave in a non-physical manner. Some level of stabilisation is therefore needed to improve the robustness of the method. Typically this is done using streamline upwind Petrov-Galerkin (SUPG) methods or enhancing a low-order element type with higher order sub-grid scale bubble functions. This chapter develops the Within-Element Petrov-Galerkin (WEPG) method that was introduced in the previous chapter. This is an upwind stabilisation method that adds numerical dissipation in to the discretisation for stability. To develop this method further, the amount of added dissipation it introduces is optimised to match analytical solutions in one-dimension. The scheme is extended into multi-dimensions and is demonstrated in a variety of test problems using different angular basis functions in direction of particle travel. [1]

[1] This chapter has been published in Annals of Nuclear Energy, (Merton et al., 2008)

Contents

3.1. Introduction

This chapter describes the development of two optimal discontinuous finite element (FE) Riemann methods and their application to the one-speed Boltzmann transport equation in the steady-state. The methods that are developed here are based on the Within-Element Petrov-Galerkin (WEPG) method introduced in chapter (2). The methods control oscillation and unphysical behaviour in the solution by adding artificial numerical dissipation in to the discretisation. This added dissipation increases the amount of upwinding in the finite element stencil which stabilises the solution by damping unwanted oscillation. The proposed methods optimise the amount of dissipation applied in the streamline direction. This dissipation is applied within an element using a novel Riemann FE method which is based on an analogy between control volume discretisation methods and finite element methods when integration by parts is applied to the transport terms. In one dimension the optimal finite element solutions match the analytical solution exactly at each outlet node, by tuning the amount of added dissipation. Both schemes couple elements in space via a Riemann approach. The first of the two schemes is a Petrov-Galerkin (PG) method which introduces dissipation via the equation residual. The second scheme uses a streamline diffusion stabilisation term in the discretisation. These two methods provide a discontinuous Petrov-Galerkin (DPG) scheme that can stabilise an element across the full range of radiation regimes, obtaining robust solutions with suppressed oscillation. Three basis functions in angle

of particle travel have been implemented in an optimal DPG Riemann solver which include the P_N (spherical harmonic), S_N (discrete ordinate) and LW_N (linear octahedral wavelet) angular expansions. These methods are applied to a series of demanding two-dimensional radiation transport problems. As noted previously, the concept of optimal discontinuous Galerkin (DG) methods that match analytical solutions is not new and has already been achieved using positive diffusion in one spatial dimension (Pain and Goddard, 2000). The current work furthers this research by extending the optimal method to multi-dimensions and multi-angle, and introduces methods for tuning the amount of added dissipation that are believed to be new. The mathematical basis of the Riemann method that the new schemes use is described in (Pain et al., 2006a).

In chapter (2), the one-speed steady-state form of the Boltzmann equation was discretised in multi-dimensional space using Galerkin-weighted linear discontinuous finite elements. This forms the basis of the optimal schemes. The angular expansion is performed before the spatial expansion, as usual when discretising the Boltzmann equation. The Riemann framework used allows any angular expansion scheme to be swapped in without having to modify the discretised equation. This has been illustrated with the implementation of P_N, S_N and the LW_N methods in the optimal finite element Riemann solver. Therefore, the work in the present chapter is completely general and not specific to any particular angular discretisation scheme. An orthogonal mesh strategy in Cartesian geometry is used in the present work. Some further development of the schemes may be required to use them on different types of meshes in other geometries. This is left as a topic of future work; only orthogonal meshes in Cartesian geometry are considered here.

The present chapter has been arranged as follows. In section (3.2) the concept of streamline upwind stabilisation using Riemann terms is introduced and consideration given to the existing PG framework that has been developed for radiation transport. In section (3.3), the condition that is imposed on the residual on the element boundary is discussed; it is set to zero in the surface terms which is a natural boundary condition of the residual form of the equations. In section (3.4) the optimal coefficients are derived that tune the amount of upwind weighting added to an element to stabilise it. Expressions for the optimal coefficients are derived for a residual-based dissipation and also for a diffusion-based dissipation. These expressions tune the amount of dissipation that is added so that the discontinuous finite element solution matches the analytical solution in one-dimension. To formulate this, the multi-dimensional system is diagonalised into a set of one-

dimensional, S_N-like waves each of which has an optimal coefficient. The diagonalisation scheme is described in section (3.5), which formulates a stabilisation matrix containing the optimal coefficient. This distributes the optimal coefficients in angle appropriately across the expansion. The consequence of adding too much dissipation is considered in section (3.6) and how this might be avoided. Adding artificial dissipation to any scheme can effect the accuracy of the discretisation. This is investigated in section (3.7) where it is shown what can be expected of the optimal schemes in terms of their convergence properties. Each scheme is expanded in a Taylor analysis to find its truncation order. It is found that in general, these methods compare favourably with other differencing schemes in terms of their spatial truncation. In section (3.8) numerical results from a series of radiation transport problems are presented for the optimal finite element schemes, demonstrating their steady-state performance. Conclusions are drawn in section (3.9) with recommendations for further work. These recommendations include adding non-linearity to control the dissipation in the direction of the solution gradient.

3.2. Petrov-Galerkin Discretisation

In the discontinuous Galerkin scheme, continuity of solution is not enforced across interfaces of elements. This permits the capture of sharp gradients and near discontinuities in the radiation field. To ensure the scheme is able to do this while remaining well behaved, and to suppress non-physical oscillations occurring a Petrov-Galerkin term is added to the standard form of the equations. Writing the angularly discretised equation (2.15) with this modification one has:

$$(\mathbf{I} - \mathbf{A}\cdot\nabla\mathbf{P})[(\mathbf{A}\cdot\nabla + \mathbf{H}(\mathbf{r}))\Psi(\mathbf{r}) - \mathbf{S}(\mathbf{r})] = \mathbf{0}, \tag{3.1}$$

where $\mathbf{P}$ is the $\mathcal{M}\times\mathcal{M}$ stabilisation matrix and $\mathcal{M}$ is the number of moments of the angular expansion. $\mathbf{0}$ is a vector of length $\mathcal{M}$ containing zeroes. $\mathbf{I}$ is the $\mathcal{M}\times\mathcal{M}$ identity matrix, and $\mathbf{H}$ is the $\mathcal{M}\times\mathcal{M}$ angularly discrete scattering-removal operator and $\mathbf{A}$ the Jacobian of the angle of particle travel. The vector $\mathbf{S}$ is the sum of all the angular discrete sources that are driving the system. Described in some detail in section (3.5.3), the choice of $\mathbf{P}$ determines the type of SUPG weighting. One may write equation (3.1) in terms of the equation residual $\mathscr{R}$

which is more compact:

$$(\mathbf{I} - \mathbf{A} \cdot \nabla \mathbf{P})\mathscr{R} = \mathbf{0}. \tag{3.2}$$

Artificial dissipation is introduced via the additional upwind term. This has the effect of improving numerical stability of the finite element differencing scheme and maintaining robustness across different radiation regimes. Alternatively, an element can be stabilised with diffusive modes in which case the modified residual form of the angularly discretised equation (3.2) is instead:

$$\mathscr{R} - \mathbf{A} \cdot \nabla \mathbf{P} \mathbf{A} \cdot \nabla \Psi(\mathbf{r}) = \mathbf{0}. \tag{3.3}$$

With the Petrov-Galerkin approach, care must be taken to prevent the stabilisation term becoming too large. If too much dissipation is introduced, more advection will occur backwards than forwards and so more particles will flow out of the element than are entering in the direction of transport, and this leads to non-physical solutions. A cut-off to prevent this occurring is considered in section (3.6.1). Performing the spatial discretisation of equation (3.1) using the same approach as previously, one obtains:

$$\begin{aligned}
\int_{V_E} dV \Bigg(& -\mathbf{A} \cdot \nabla \mathbf{N}_i \sum_j \mathbf{N}_j \Psi_j + \mathbf{N}_i \mathbf{H} \sum_j \mathbf{N}_j \Psi_j - \sum_j \mathbf{N}_j s_j \\
& + \mathbf{A} \cdot \nabla \mathbf{N}_i \mathbf{P} (\mathbf{A} \cdot \nabla \sum_j \mathbf{N}_j \Psi_j + \mathbf{H} \sum_j \mathbf{N}_j \Psi_j - \sum_j \mathbf{N}_j s_j) \Bigg) \\
& + \oint_{\Gamma_E} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{N}_i \Psi_{IN} + \oint_{\Gamma_E} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{N}_i \Psi \\
& + \oint_{\Gamma_E} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{N}_i \mathscr{R} = \mathbf{0}.
\end{aligned} \tag{3.4}$$

When Green's theorem is applied to the second order derivative, a surface term containing the equation residual $\mathscr{R}$ results. Elliptic DG methods exist that spatially difference these surface integrals (Sherwin et al., 2006), but in the present work the residual has been set $\mathscr{R} = \mathbf{0}$ on the surface of the element. Note that $\mathscr{R} = \mathbf{0}$ is in fact a solution to equation (3.2), and thus provides part of a natural boundary condition of the equation. This is discussed in section (3.3).

3.3. Sub-Grid Scale Application of Boundary Conditions

With any numerical scheme that is spatially discretised on a computational mesh there is a sub-grid scale aspect to the boundary conditions. In formulating the discretisation in equation (3.4), the equation residual $\mathscr{R}$ was set to zero on the inbound surface integral. This provides a natural boundary condition for the incoming flux at the surface of each element. One may express this as:

$$\mathbf{A}\cdot\mathbf{n}\mathbf{P}\mathscr{R}+\mathbf{A}\cdot\mathbf{n}(\Psi-\Psi_{IN})=\mathbf{0}, \tag{3.5}$$

in which Ψ_{IN} is the incoming information. For the outbound flux, the boundary condition on the outgoing surface is similarly:

$$\mathbf{A}\cdot\mathbf{n}\mathbf{P}\mathscr{R}=\mathbf{0}. \tag{3.6}$$

This has a physical analogue with other areas of physics, notably with unresolved thermal boundary layers in convection problems. In one-dimensional convection problems, temperature $T(x)$ is described by the differential equation:

$$k\frac{\partial T(x)}{\partial x}-\beta(T(x)-T_0)=\mathbf{0}, \tag{3.7}$$

in which k is the thermal conductivity and β the thermal transfer coefficient. T_0 is the temperature on the boundary. In void regions this equation has the same form as the Boltzmann transport equation, and as $T(x)\longrightarrow T_0$ one may note from equation (3.7) that $\frac{\partial T(x)}{\partial x}\longrightarrow 0$. On the boundary, $\frac{\partial T(x)}{\partial x}=0$. This is analogous to the boundary condition $\mathscr{R}=\mathbf{0}$ used in the present work.

3.4. Stabilising with Optimal Dissipation

In this section, coefficient α is introduced that scales the stabilisation term. Two choices for α are derived that optimise the amount of weighting, using two complementary approaches. In section (3.5), the multi-dimensional system is diagonalised along the direction of the streamline via a Cholesky-Riemann transform. This derives a one-dimensional wave along the direction of the streamline, forming a one-dimension diagonal (S_N - like) system from the original multi-dimensional non-diagonal system. This allows each optimal stabilisation scheme to be based on one-dimensional considerations. In section (3.5.3) a strategy is developed for

distributing the optimal coefficients in angle. This involves stabilising the system while in diagonal form in Cholesky-Riemann space, and then reversing the transform to map the stabilised system back to original space. A stabilisation matrix $\mathbf{P}$ is derived for each optimal finite element Riemann method. This stabilisation matrix contains the optimal coefficient α distributed in angle. The two complementary expressions derived for α in the present section match the finite element solution at the outlet node of an element with the analytical solution exactly in one dimension. In the first case (section (3.4.1)) a true Petrov-Galerkin stabilisation of the steady-state advection problem is optimised, in which added dissipation brings in the equation residual. In the second case (section (3.4.2)), the analysis is repeated for a diffusion-stabilised scheme in which dissipation brings in purely diffusive modes.

3.4.1. DPG Method

Stabilising the steady-state advection problem with a Petrov-Galerkin term scaled with the optimal coefficient we have, based on one-dimensional considerations:

$$\int_0^l dV \mathbf{N}_i(x) \left(\mathbf{I} - \alpha l \Omega_x \frac{\partial}{\partial x} \right) \mathscr{R} = \mathbf{0}, \tag{3.8}$$

in which l is the element length and α the optimal coefficient for which an expression is required. α will describe how much dissipation is to be added to the system. $\mathbf{N}_i(x)$ is a finite element nodal basis function in the domain $0 \leq x \leq l$. Choosing l = 1, $\mathbf{N}_1(x)$ = 1-x and $\mathbf{N}_2(x)$ = x. Expanding this equation using a Galerkin projection of $\psi(x) = \sum_j \mathbf{N}_j(x)\Psi_j$ and applying Green's theorem to the advection terms one obtains:

$$\begin{aligned} \int_0^l dV \Bigg(& -\mathbf{N}_i(x) \sum_j \mathbf{N}_j(x)\Psi_j + \mathbf{N}_i(x)\sigma \sum_j \mathbf{N}_j(x)\Psi_j \\ & + \alpha \mathbf{N}_{i_x}(x) \left(\sum_j \mathbf{N}_{j_x}(x)\Psi_j \right. \\ & \left. + \sigma \sum_j \mathbf{N}_j(x)\Psi_j \right) \Bigg) + \oint_\Gamma d\Gamma \mathbf{N}_i(x) n_x \Psi(x) = \mathbf{0}. \end{aligned} \tag{3.9}$$

Computing the integrals of $\mathbf{N}_{i_x}(x)\mathbf{N}_j(x)$, $\mathbf{N}_i(x)\mathbf{N}_j(x)$ and $\mathbf{N}_{i_x}(x)\mathbf{N}_{j_x}(x)$ numerically through the domain of x the following matrix equation is obtained:

$$\left[\frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} + \sigma\begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} + \alpha\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \frac{\alpha\sigma}{2}\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}\right]\begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{3.10}$$

The inward boundary condition Ψ_{IN} has been set = 1. This choice is arbitrary, since varying it will vary the solution only at the element inlet node. The present work is concerned with matching the numerical finite element solution with the analytical solution only at the outlet node (Ψ_2). To accomplish this, Ψ_2 is set equal to the analytical solution at the outlet node, that is to substitute $\Psi_2 = e^{-\hat{\sigma}}$ into the above expression, where $\hat{\sigma} = \frac{\sigma l}{|\Omega_x|}$ is the dimensionless cross-section. Using Cramer's rule, one obtains the dimensionless expression:

$$(\frac{1}{2} - \frac{\hat{\sigma}}{6} + \alpha - \frac{\alpha\hat{\sigma}}{2}) = (\frac{1}{2} + \frac{\hat{\sigma}}{3} + \alpha + \frac{\hat{\sigma}^2}{12} + \frac{\alpha\hat{\sigma}}{2})e^{-\hat{\sigma}}, \tag{3.11}$$

which can be arranged in terms of α, the optimal coefficient, to obtain the final expression:

$$\alpha = \frac{(6 + 4\hat{\sigma} + \hat{\sigma}^2) + (2\hat{\sigma} - 6)e^{\hat{\sigma}}}{(12 - 6\hat{\sigma})e^{\hat{\sigma}} - (12 + 6\hat{\sigma})}. \tag{3.12}$$

When this optimal coefficient is placed into the discretised equation, one should use a length scale of $\frac{l}{|\Omega_x|}$ to ensure the dissipation is introduced across the length of the element seen by the ray along direction Ω_x. This will then match the finite element solution with the exact solution, for the complete set of one-dimensional waves in the quadrature set. The graph of figure (3.1) shows how the optimal amount of dissipation varies with material. Figure (3.2) demonstrates this form for α matches the analytical solution at the outlet node of an element for a simple test case.

3.4.2. Diffusion Method

It has been suggested that where the transport is dominated by strong diffusive modes, a diffusion-based stabilisation may be more appropriate and could work better than a full Petrov-Galerkin stabilisation such as that described in section

(3.4.1). This section develops an expression for coefficient α based on a dissipation that is purely diffusive. Stabilising the steady-state advection problem in this manner, again based on one-dimensional considerations:

$$\int_0^l dV \mathbf{N}_i(x) \left(\mathscr{R} - \alpha l \Omega_x \frac{\partial^2 \Psi(x)}{\partial x^2} \right) = \mathbf{0}, \tag{3.13}$$

in which l is the element scale and α the optimal coefficient for which an expression is required. α will describe how much negative diffusion is to be added to the system. Choosing the same set of nodal basis functions $\mathbf{N}_i(x)$ as those used in section (3.4.1), this equation is expanded using a Galerkin projection of Ψ_x. Application of Green's theorem to the advection terms then obtains:

$$\begin{aligned} \int_0^l dV \Bigg(& -\mathbf{N}_{i_x}(x) \sum_j \mathbf{N}_j(x)\psi_j + \mathbf{N}_i(x)\sigma \sum_j \mathbf{N}_j(x)\psi_j \\ & + \alpha \mathbf{N}_{i_x}(x) \sum_j \mathbf{N}_{j_x}(x)\psi_j \Bigg) + \oint_\Gamma d\Gamma \mathbf{N}_i(x) n_x \Psi(x) = \mathbf{0}. \end{aligned} \tag{3.14}$$

Computing the integrals of $\mathbf{N}_{i_x}(x)\mathbf{N}_j(x)$, $\mathbf{N}_i(x)\mathbf{N}_j(x)$ and $\mathbf{N}_{i_x}(x)\mathbf{N}_{j_x}(x)$ numerically one obtains the matrix equation:

$$\begin{aligned} & \left[\frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} + \sigma \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix} \right. \\ & \left. + \alpha \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \right] \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix} = \begin{pmatrix} \Psi_{IN} \\ 0. \end{pmatrix}, \end{aligned} \tag{3.15}$$

which has the form $\mathbf{Ax} = \mathbf{b}$. As with the full Petrov-Galerkin case, the inward boundary condition Ψ_{IN} has been set = 1. Setting the outbound solution Ψ_2 equal to the analytical solution, using the same definition of $\hat{\sigma}$ as in the full Petrov-Galerkin case one obtains using Cramer's rule:

$$(\frac{1}{2} - \frac{\hat{\sigma}}{6} + \alpha) = (\frac{1}{2} + \frac{\hat{\sigma}^2}{12} + \alpha\hat{\sigma} + \frac{\hat{\sigma}}{3} + \alpha)e^{-\hat{\sigma}}, \tag{3.16}$$

which may be arranged in terms of α, the optimal coefficient, to obtain a dimensionless expression for the diffusion case:

$$\alpha = \frac{(6 + 4\hat{\sigma} + \hat{\sigma}^2) - (6 - 2\hat{\sigma})e^{\hat{\sigma}}}{12e^{\hat{\sigma}} - 12\hat{\sigma} - 12}. \tag{3.17}$$

As in the previous case, this expression is extended to multi-angle by choosing a length scale $h = \frac{l}{|\Omega_x|}$. This matches the finite element solution at the outlet node with the exact solution to a set of one-dimensional rays.

3.5. Optimal Coefficient in Riemann Space

In the previous section, two expressions for optimal coefficient α were derived that match the analytical solution exactly for a set of one-dimensional diagonal (S_N-like) waves. However, diagonal waves exist only for diagonal systems of equations such as the S_N equations. The P_N equations are not diagonal in normal space, and for both optimal methods to be general they must be applied in a space where all systems are diagonal. In section (3.5.1), a Cholesky-Riemann transform is developed that diagonalises the transport equation regardless of how it were discretised in angle. When one applies this Cholesky-Riemann transform to the one-dimensional P_N equations, one obtains the one-dimensional S_N equations on a Legendre quadrature set[2]. In the diagonal Cholesky-Riemann space, the solutions to this are the angular fluxes and their combination into a scalar flux (meaningful only in the diagonal space) will produce a ray-effect. If the angular fluxes are mapped back to the original space by reversing the transform, they become angular moments and form the ray-free P_N solutions one would obtain from solving the original (non-diagonal) equations in normal space. When the S_N equations are decomposed in this manner however, they remain unaltered since they were diagonal to begin with. In order to apply α to each moment correctly, it is necessary to distribute it appropriately in angle across the expansion. A strategy is developed in section (3.5.2) that introduces the optimal dissipation into the equations after they have been mapped in to the diagonal Cholesky-Riemann space, stabilising

[2]This raises the possibility of solving the P_N equations using conventional sweep-based marching algorithms designed for S_N. Unfortunately, the P_N equations are not susceptible to this approach in multi-dimensions, since each directional term in the transport equation will be diagonal in a different space. It is not possible to have all x, y, z terms diagonal in the same space simultaneously, preventing a fully diagonal system from being formed in multi-dimensions. To use the optimal methods with P_N in multi-dimensions, the diagonalisation is performed along the streamline. This produces a one-dimensional diagonal system in Cholesky-Riemann space

the diagonal form of the equations. In diagonal form in Cholesky-Riemann space, the system is a one-dimensional problem along the direction of the streamline, and so the one-dimensional definitions of α apply. First, the one-dimensional diagonal system is derived from the multi-dimensional non-diagonal system. Then coefficient α is inserted to achieve an optimal stability in the diagonal system. Once a stabilised equation has been obtained here the diagonalisation is reversed. Reversing the diagonalisation maps down to original variables with the effect of appropriately distributing the added dissipation in angle across the expansion.

3.5.1. Diagonalising the System

The angularly discretised steady-state form of the transport equation for direction n in the presence of no applied source, may be written:

$$\mathbf{A}_n \frac{\partial \Psi}{\partial n} + \mathbf{H}\Psi = \mathbf{0}, \tag{3.18}$$

where direction n is arbitrary. $\mathbf{H}$ is the scattering-removal operator, Ψ the vector of angular unknowns and $\mathbf{A}_n$ the angular Jacobian associated with dimension n. The Riemann decomposition of angular Jacobian matrix $\mathbf{A}_n = \mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n$ and of $\mathbf{H} = \mathbf{R}_n \mathbf{H}_n \mathbf{L}_n$ is obtained, where $\mathbf{L}_n$ and $\mathbf{R}_n$ are the left and right eigenvectors of $\mathbf{A}_n$ respectively, and $\mathbf{\Lambda}_n$ the diagonal matrix of eigenvalues associated with the Riemann decomposition of direction n. $\mathbf{H}_n$ is a diagonal matrix in Riemann space associated with $\mathbf{H}$ in normal space. One can therefore write:

$$\mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n \frac{\partial \Psi}{\partial n} + \mathbf{R}_n \mathbf{H}_n \mathbf{L}_n \Psi = \mathbf{0}. \tag{3.19}$$

Using the similarity transform $\mathbf{L}_n \Psi = \Psi_n$ to map original variable Ψ up to Riemann space:

$$\mathbf{R}_n \mathbf{\Lambda}_n \frac{\partial \Psi_n}{\partial n} + \mathbf{R}_n \mathbf{H}_n \Psi_n, \tag{3.20}$$

where Ψ_n and $\mathbf{H}_n$ are referred to as characteristic variables. In order to complete the diagonalisation of the system, the Cholesky factorisation of $\mathbf{H}_n = \mathbf{G}^T \mathbf{G}$ is inserted. Inserting the Cholesky factorisation of $\mathbf{H}_n$ in equation (3.20) and multiplying by the left eigenvector $\mathbf{L}_n$ one obtains:

$$\mathbf{\Lambda}_n \frac{\partial \Psi_n}{\partial n} + \mathbf{G}^T \mathbf{G} \Psi_n = \mathbf{0}, \tag{3.21}$$

where the property $\mathbf{L}_n\mathbf{R}_n = \mathbf{R}_n^{-1}\mathbf{R}_n = \mathbf{I}$ has been used. Matrix $\mathbf{H}$ is diagonal typically only for S_N and P_N expansions. For wavelet expansions it is not diagonal. Thus the Riemann transform alone is insufficient to guarantee a completely diagonal system. In order to restore the diagonal system one must apply this Cholesky factorisation. This operation is only possible if $\mathbf{H}_n$ is a positive definite matrix. In radiation transport processes it is always the case that $\mathbf{H}_n$ is positive definite. Multiplying through by $\mathbf{G}^{-T}$ and making the substitution $\Psi_n = \mathbf{G}^{-1}\Psi_n^{'}$:

$$\mathbf{G}^{-T}(\mathbf{\Lambda}_n\mathbf{G}^{-1}\frac{\partial\Psi_n^{'}}{\partial n} + \mathbf{G}^T\Psi_n^{'}) = \mathbf{0}, \tag{3.22}$$

where the property $\mathbf{G}\mathbf{G}^{-1} = \mathbf{I}$ has been used. Defining a matrix $\mathbf{A_G} = \mathbf{G}^{-T}\mathbf{\Lambda}_n\mathbf{G}^{-1}$ and inserting it into equation (3.22):

$$\mathbf{A_G}\frac{\partial\Psi_n^{'}}{\partial n} + \mathbf{I}\Psi_n^{'} = \mathbf{0}. \tag{3.23}$$

The Riemann decomposition of $\mathbf{A_G}$ is now obtained in which $\mathbf{R_G}$ and $\mathbf{L_G}$ are the right and left eigenvectors of $\mathbf{A_G}$ respectively and $\mathbf{\Lambda_G}$ is a diagonal matrix of eigenvalues associated the Cholesky-Riemann decomposition. Inserting $\mathbf{A_G} = \mathbf{R_G}\mathbf{\Lambda_G}\mathbf{L}_G$ into equation (3.23):

$$\mathbf{R_G}\mathbf{\Lambda_G}\mathbf{L_G}\frac{\partial\Psi_n^{'}}{\partial n} + \mathbf{I}\Psi_n^{'} = \mathbf{0}. \tag{3.24}$$

Substituting $\mathbf{L_G}\Psi_n^{'} = \Psi_n^{''}$ in the first term and $\Psi_n^{'} = \mathbf{R_G}\Psi_n^{''}$ in the second in order to map both instances of $\Psi_n^{'}$ into the same space as eigenvalue $\mathbf{\Lambda_G}$, one obtains:

$$\mathbf{R_G}\mathbf{\Lambda_G}\frac{\partial\Psi_n^{''}}{\partial n} + \mathbf{R_G}\Psi_n^{''} = \mathbf{0}. \tag{3.25}$$

Finally, multiplying this expression by $\mathbf{L_G} = \mathbf{R_G^{-1}}$ obtains the following diagonal (S_N-like) set of equations, irrespective of the angular expansion:

$$\mathbf{\Lambda_G}\frac{\partial\Psi_n^{''}}{\partial n} + \mathbf{I}\Psi_n^{''} = \mathbf{0}. \tag{3.26}$$

Clearly, any solutions to this will be meaningful only in diagonal Cholesky-Riemann space, although if mapped back through the diagonalisation precisely, they would match solutions to the original system obtained in the original space.

3.5.2. Stabilising the Diagonal System

Equation (3.26) may be discretised in the diagonal space using each optimal method for α. Applying an optimal amount of diffusion to the diagonal system in direction n one obtains:

$$\mathbf{\Lambda_G}\frac{\partial \Psi_n^{''}}{\partial n} + \mathbf{I}\Psi_n^{''} - \mathbf{\Lambda_{G_\alpha}}\frac{\partial^2 \Psi_n^{''}}{\partial n^2} = \mathbf{0}, \tag{3.27}$$

where $\mathbf{\Lambda_{G_\alpha}}$ is a diagonal matrix that optimises the artificial diffusion term along direction n. The coefficient for moment μ is at position μ along the main diagonal:

$$\mathbf{\Lambda_{G_{\alpha_{\mu\nu}}}} = \begin{cases} h|\mathbf{\Lambda_{G_{\mu\nu}}}|\alpha(\hat{\sigma}) & \mu = \nu; \\ \mathbf{0} & \mu \neq \nu, \end{cases} \tag{3.28}$$

in which $\hat{\sigma} = \frac{\mathbf{I}h}{|\mathbf{\Lambda_{G_{\mu\nu}}}|}$ is the dimensionless cross-section where h is the length of the element and μ and ν are moment number. $\alpha(\hat{\sigma})$ is obtained using the one-dimensional derivation for the optimal diffusion coefficient. Operating with $|\ \mathbf{\Lambda_{G_{\mu\nu}}}\ |$ thus produces a diagonal matrix of optimal coefficients each tuned to a particular moment in the expansion. The Cholesky-Riemann map has decoupled the problem into a separate wave along direction n. Therefore the one-dimensional definition of α is valid here for all n, and will match the exact solution in diagonal space for each ray in the expansion of $\Psi_n^{''}$. It is required that $\Psi_n^{''}$ be mapped back to original variable Ψ so that one can proceed with the spatial discretisation. In order to map back to original variables one must precisely reverse the diagonalisation. One begins by reversing step (3.26), multiplying by $\mathbf{R_G}$ to obtain:

$$\mathbf{R_G\Lambda_G}\frac{\partial \Psi_n^{''}}{\partial n} + \mathbf{R_G}\Psi_n^{''} - \mathbf{R_G\Lambda_{G_\alpha}}\frac{\partial^2 \Psi_n^{''}}{\partial n^2}. \tag{3.29}$$

Next, the substitution $\Psi_n^{''} = \mathbf{L_G}\Psi_n^{'}$ is made in both the DG term and the added diffusion term. This operation obtains an α-modified form of equation (3.24) that uses an optimal amount of diffusion for stability:

$$\mathbf{R_G\Lambda_G L_G}\frac{\partial \Psi_n^{'}}{\partial n} + \mathbf{R_G L_G}\Psi_n^{'} - \mathbf{R_G\Lambda_{G_\alpha} L_G}\frac{\partial^2 \Psi_n^{'}}{\partial n^2} = \mathbf{0}. \tag{3.30}$$

Next, the first and second order terms are mapped down from Cholesky-Riemann space in the transforms $\mathbf{A_G} = \mathbf{R_G\Lambda_G L_G}$ and $\mathbf{A_{G_\alpha}} = \mathbf{R_G\Lambda_{G_\alpha} L_G}$. This re-

verses step (3.24) and one obtains:

$$\mathbf{A_G}\frac{\partial \Psi_n^{'}}{\partial n} + \mathbf{I}\Psi_n^{'} - \mathbf{A_{G_\alpha}}\frac{\partial^2 \Psi_n^{'}}{\partial n^2} = \mathbf{0}, \tag{3.31}$$

in which $\mathbf{A_{G_\alpha}}$ is an angular Jacobian that distributes the diagonal $\mathbf{\Lambda_{G_\alpha}}$ matrix in angle across the expansion. Applying α in Cholesky-Riemann space in this manner customises the dissipation for each moment, acting on oscillation in the higher moments, where other methods may have no effect. One proceeds by reversing step 3.23 which involves decomposing angular Jacobians $\mathbf{A_G}$ and $\mathbf{A_{G_\alpha}}$ in terms of their Cholesky factorisation. Thus the substitutions $\mathbf{A_G} = \mathbf{G}^{-T}\mathbf{\Lambda}_n\mathbf{G}^{-1}$ and $\mathbf{A_{G_\alpha}} = \mathbf{G}^{-T}\mathbf{\Lambda}_{n_\alpha}\mathbf{G}^{-1}$ in equation (3.31) are made. Note that this re-introduces the diagonal matrix of eigenvalues $\mathbf{\Lambda}_n$ into direction n. One obtains:

$$\mathbf{G}^{-T}\mathbf{\Lambda}_n\mathbf{G}^{-1}\frac{\partial \Psi_n^{'}}{\partial n} + \mathbf{I}\Psi_n^{'} - \mathbf{G}^{-T}\mathbf{\Lambda}_{n_\alpha}\mathbf{G}^{-1}\frac{\partial^2 \Psi_n^{'}}{\partial n^2} = \mathbf{0}, \tag{3.32}$$

and reversing step (3.22) by making the substitution $\Psi_n^{'} = \mathbf{G}\Psi_n$:

$$\mathbf{\Lambda}_n\frac{\partial \Psi_n}{\partial n} + \mathbf{G}^T\mathbf{G}\Psi_n - \mathbf{\Lambda}_{n_\alpha}\frac{\partial^2 \Psi_n}{\partial n^2} = \mathbf{0}, \tag{3.33}$$

and multiplying through by $\mathbf{G}^T$. Reversing the Cholesky factorisation of $\mathbf{H}_n$ and multiplying through by Riemann variable $\mathbf{R}_n$ step (3.21) is reversed to obtain:

$$\mathbf{R}_n\mathbf{\Lambda}_n\frac{\partial \Psi_n}{\partial n} + \mathbf{R}_n\mathbf{H}_n\Psi_n - \mathbf{R}_n\mathbf{\Lambda}_{n_\alpha}\frac{\partial^2 \Psi_n}{\partial n^2} = \mathbf{0}. \tag{3.34}$$

Using $\Psi_n = \mathbf{L}_n\Psi$ in each term step (3.20) is reversed, completing the map down of $\Psi_n^{''}$ to original variable Ψ, obtaining:

$$\mathbf{R}_n\mathbf{\Lambda}_n\mathbf{L}_n\frac{\partial \Psi}{\partial n} + \mathbf{R}_n\mathbf{H}_n\mathbf{L}_n\Psi - \mathbf{R}_n\mathbf{\Lambda}_{n_\alpha}\mathbf{L}_n\frac{\partial^2 \Psi}{\partial n^2} = \mathbf{0}, \tag{3.35}$$

which is a form of equation (3.19), but consistent with the modifications for optimal stability via the second order term. To properly distribute the optimal stability in angle about each moment one completes the map-down from Riemann space using the transforms $\mathbf{A}_n = \mathbf{R}_n\mathbf{\Lambda}_n\mathbf{L}_n$, $\mathbf{H} = \mathbf{R}_n\mathbf{H}_n\mathbf{L}_n$ and $\mathbf{A}_{n_\alpha} = \mathbf{R}_n\mathbf{\Lambda}_{n_\alpha}\mathbf{L}_n$ and obtains:

$$\mathbf{A}_n\frac{\partial \Psi}{\partial n} + \mathbf{H}\Psi - \mathbf{A}_{n_\alpha}\frac{\partial^2 \Psi}{\partial n^2} = \mathbf{0}. \tag{3.36}$$

This reverses step (3.19), obtaining a form of equation (3.18) with the modifications for optimal stability in direction n. The optimal discontinuous Petrov-Galerkin treatment follows along a similar route.

3.5.3. Optimal Stabilisation Matrix

In this section, an expression for the stabilisation matrix $\mathbf{P}$ is derived for the optimal finite element Riemann methods. A general form for the angularly discretised Boltzmann equation modified for stability with an upwind term is given in equation (3.1). The modification to this in order to implement the optimal Riemann methods is to use a $\mathbf{P}$ that has the stability coefficients distributed in angle around it. Equating the upwind term in equation (3.1) with that in equation (3.36) one obtains a matrix expression for $\mathbf{P}$:

$$\mathbf{A}_n \mathbf{P} \mathbf{A}_n = \mathbf{A}_{n_\alpha} = \mathbf{R}_n \mathbf{G}^T \mathbf{R}_\mathbf{G} \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{L}_\mathbf{G} \mathbf{G} \mathbf{L}_n. \tag{3.37}$$

It is not appropriate to invert $\mathbf{A}_n$, since it has zero entries on the main diagonal. In any case, it is along the streamline direction. In order to obtain $\mathbf{P}$, 3.37 is expressed in terms of the appropriate Riemann decomposition:

$$\mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n \mathbf{P} \mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n = \mathbf{R}_n \mathbf{G}^T \mathbf{R}_\mathbf{G} \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{L}_\mathbf{G} \mathbf{G} \mathbf{L}_n. \tag{3.38}$$

Multiplying each side of this expression by $\mathbf{L}_n$ and then operating on $\mathbf{R}_n$ with each side one obtains:

$$\mathbf{\Lambda}_n \mathbf{L}_n \mathbf{P} \mathbf{R}_n \mathbf{\Lambda}_n = \mathbf{G}^T \mathbf{R}_\mathbf{G} \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{L}_\mathbf{G} \mathbf{G}. \tag{3.39}$$

Operating on each side with $\mathbf{\Lambda}_n^{-1}$ and then a second time again with $\mathbf{\Lambda}_n^{-1}$, one obtains:

$$\mathbf{L}_n \mathbf{P} \mathbf{R}_n = \mathbf{\Lambda}_n^{-1} \mathbf{G}^T \mathbf{R}_\mathbf{G} \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{L}_\mathbf{G} \mathbf{G} \mathbf{\Lambda}_n^{-1}. \tag{3.40}$$

Operating on each side with $\mathbf{R}_n$ and then operating on $\mathbf{L}_n$ with each side one obtains:

$$\mathbf{P} = \mathbf{R}_n \mathbf{\Lambda}_n^{-1} \mathbf{G}^T \mathbf{R}_\mathbf{G} \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{L}_\mathbf{G} \mathbf{G} \mathbf{\Lambda}_n^{-1} \mathbf{L}_n. \tag{3.41}$$

Once $\mathbf{P}$ has been formed in this manner it is inserted into the discretised equation, correctly upwinding the dissipation. This approach appropriately distributes the stability coefficient α in angle about each moment in the expansion.

3.6. Streamline Upwind Stabilisation Using the Finite Element Riemann Method

In the current work, dissipation is applied within an element using a novel Riemann FE method which is based on an analogy between control volume discretisation methods and finite element methods when integration by parts is applied to the transport terms. Integrating equation (3.4) by parts and applying Green's theorem to the advection terms produces a term $\mathbf{A} \cdot \nabla \mathbf{N}_i$. Any dissipation applied within an element can therefore act only along the direction of $\nabla \mathbf{N}_i$. The Jacobian $\mathbf{A}_n$ that was used in section (3.5.1) when diagonalising the system, and also in equation (3.37) when deriving matrix $\mathbf{P}$ is defined according to $\mathbf{A}_{\hat{n}} = n_x \mathbf{A}_x + n_y \mathbf{A}_y + n_z \mathbf{A}_z$. In this definition, n_x, n_y and n_z are defined similarly to how face normals would be defined in a control volume method. These are determined according to $n_x = \frac{1}{|\nabla \mathbf{N}_i|} \frac{\partial \mathbf{N}_i}{\partial x}$, $n_y = \frac{1}{|\nabla \mathbf{N}_i|} \frac{\partial \mathbf{N}_i}{\partial y}$ and $n_z = \frac{1}{|\nabla \mathbf{N}_i|} \frac{\partial N_i}{\partial z}$ and thus direct the dissipation along the $\nabla \mathbf{N}_i$ term. The choice of $\mathbf{P}$ defined in section (3.5.3) is therefore similar to that used in control-volume implementations of Riemann methods, where an angular Jacobian $\mathbf{A}_n = n_x \mathbf{A}_x + n_y \mathbf{A}_y + n_z \mathbf{A}_z$ would be used in equation (3.4).

3.6.1. Excessive Streamline Dissipation

Material cross-sections may reach values which give rise to SUPG weightings large enough to dissipate more particles than entered the element. This is known as backward advection and will cause non-physical solutions in the particle flux. A cut-off should therefore be provided that makes some attempt to preclude this. In this section, such a cut-off is defined. The optimal coefficient α derived in section (3.4.1) must not exceed this if backward advection is to be averted. The SUPG term in equation (3.1) imports a second first-order advection term when it operates on the material term ($\mathbf{H}\Psi$) of the equation residual $\mathscr{R}$. This second advection term is in the opposite direction to the transport since it is a dissipation, and may result in more particles flowing backwards across an element than forwards if the SUPG weighting becomes large enough. Such non-physical scenarios can result

in the stabilisation properties of the SUPG scheme breaking down, perhaps generating oscillation absent in the unstabilised (standard DG) scheme. Thus a cut-off in the amount of SUPG weighting is needed in order to avert excess dissipation being introduced to the system. If the cut-off is exceeded the SUPG term should be scaled out, in such a way as to be recovered again when it is needed or when local behaviour of solution improves. There will be no net transport to first order if the condition on the first order advection terms:

$$\mathbf{\Lambda}_{\mathbf{G}} \frac{\partial \Psi_n''}{\partial n} - \mathbf{\Lambda}_{\mathbf{G}_\alpha} \mathbf{I} \frac{\partial \Psi_n''}{\partial n} = \mathbf{0}, \tag{3.42}$$

is met in diagonal space. This is because the number of particles that are flowing backwards as a result of the dissipation introduced to the diagonal system is balanced by the number of particles that are entering the element in the direction of transport. Note that in diagonal space the scattering removal operator $\mathbf{H}$ is always the identity matrix $\mathbf{I}$, and is thus guaranteed to be spatially invariant. It follows from this expression that if $\mathbf{\Lambda}_{\mathbf{G}_{\alpha_{\mu\nu}}} \mathbf{I}_{\mu\nu} > \mathbf{\Lambda}_{\mathbf{G}_{\mu\nu}}$ then more particles will flow backwards than forwards along ray μ in diagonal space, and the SUPG term should be scaled out. Inserting the definition of $\mathbf{\Lambda}_{\mathbf{G}_\alpha}$ from section (3.5.2):

$$\mathbf{\Lambda}_{\mathbf{G}_{\mu\nu}} - \frac{\mathbf{\Lambda}_{\mathbf{G}_{\mu\nu}} h\alpha}{|\mathbf{\Lambda}_{\mathbf{G}_{\mu\nu}}|} \mathbf{I}_{\mu\nu} = \mathbf{0}, \tag{3.43}$$

which gives rise to the following cut-off value for the optimal coefficient in the diagonal space, along ray μ:

$$\alpha > \frac{|\mathbf{\Lambda}_{\mathbf{G}_{\mu\mu}}|}{h}, \tag{3.44}$$

which generalises in all other spaces to:

$$\alpha > \frac{|\mathbf{\Lambda}_{\mathbf{G}_{\mu\mu}}|}{h\mathbf{H}_{\mu\mu}}. \tag{3.45}$$

3.7. Error Analysis

Truncation error, consistency, stability and convergence are several concepts from numerical analysis that describe the fidelity and overall quality of a spatial discretisation scheme. This section is concerned with the spatial truncation error of the difference schemes discussed in this section. There is a computational efficiency

argument in favour of difference schemes with high order spatial truncation, since coarser grid sizes achieve more acceptable difference solutions. For example, it is well known that central difference schemes exhibit second order truncation. It is expected the standard discontinuous Galerkin scheme exhibit third order at the outlet node, but perhaps only second order at the inlet due to the full upwinding of the incoming information on this boundary. It is expected the two optimal methods will exhibit lower order than the standard DG method, due to the artificial dissipation they introduce. The truncation order for diamond difference, standard DG and the two optimal methods have been obtained and the results are shown in table (3.1). To obtain these results, an algebraic solution was derived for each scheme. This expressed the outbound flux in terms of the incoming boundary condition and the element width. The algebraic solution was then expanded in a Taylor series and compared with the Taylor series of the analytical solution, equation (3.46). The order up to which the series agree suggests the truncation order of the scheme. This has been demonstrated in detail for the optimal DPG method, and the analysis for the other methods follows along a similar route. Based on one-dimensional considerations, the flux Ψ_{out} flowing out from an element of length Δ may be obtained analytically in terms of the flux Ψ_{in} entering the element, to be:

$$\frac{\Psi_{out}}{\Psi_{in}} = e^{-\sigma\Delta} = e^{-2h}, \tag{3.46}$$

where σ is the macroscopic total cross-section of the material inside the element and $h = \frac{\sigma\Delta}{2}$. For the optimal DPG scheme, the algebraic solution that expresses Ψ_{out} in terms of Ψ_{in} and h is found to be:

$$\frac{\Psi_{out}}{\Psi_{in}} = \frac{6-5h}{6+7h+2h^2}, \tag{3.47}$$

where the substitution $h = \frac{\sigma\Delta}{2}$ has again been made. Similar expressions for the other methods are readily obtained. These are tabulated in table (3.1). From equation (3.46), when the optimal DPG scheme is applied the following approximation is thus being made:

$$\frac{6-5h}{6+7h+2h^2} \approx e^{-2h}. \tag{3.48}$$

It is noted from equation (3.47) that elements with $h > \frac{6}{5}$ will produce negative Ψ_{out} with this method. The maximum value of h that averts negative Ψ_{out} is denoted here as h_{max}. It is found that h_{max} is quite different for each scheme. The different values of h_{max} are tabulated in table (3.1). Expanding the left hand side of equation (3.48) in a Taylor series, one can readily obtain:

$$\frac{6-5h}{2h^2+7h+6} = 1 - 2h + 2h^2 - \frac{5h^3}{3} + ... \tag{3.49}$$

Expanding the right hand side of equation (3.48) in a Taylor series, one can obtain:

$$e^{-2h} = 1 - 2h + \frac{4h^2}{2!} - \frac{8h^3}{3!} + \frac{16h^4}{4!} - \frac{32h^5}{5!} + ... \tag{3.50}$$

Inspecting series (3.49) and (3.50) it is seen that they are identical up to $O(h^2)$ and so:

$$\frac{6-5h}{6+7h+2h^2} = O(h^2)e^{-2h}. \tag{3.51}$$

Therefore the approximation (3.48) is exact up to $O(h^2)$. Hence the optimal DPG scheme is at most second order accurate in terms of spatial truncation error. The Taylor analysis for the other schemes follows in a similar manner to that illustrated here for optimal DPG. The truncation orders are tabulated in table (3.1) for each scheme.

Scheme	$\frac{\Psi_{out}}{\Psi_{in}}$	**Taylor Expansion**	**O**	$\mathbf{h_{max}}$
Analytical	e^{-2h}	$1 - 2h + \frac{4h^2}{2!} - \frac{8h^3}{3!} + \cdots$	-	-
Diamond	$\frac{1-h}{1+h}$	$1 - 2h + 2h^2 - 2h^3 + 2h^4 - \cdots$	2	1
Standard DG	$\frac{3-2h}{3+4h+2h^2}$	$1 - 2h + 2h^2 + \frac{4h^3}{3} + \frac{4h^4}{9} + \cdots$	3	$\frac{3}{2}$
Optimal DPG	$\frac{6-5h}{6+7h+2h^2}$	$1 - 2h + 2h^2 - \frac{5h^3}{3} + \frac{23h^4}{18} - \cdots$	2	$\frac{6}{5}$
Optimal Diff.	$\frac{3-h}{3+5h+h^2}$	$1 - 2h + 3h^2 - \frac{13h^3}{3} + \frac{56h^4}{9} - \cdots$	1	3

Table 3.1.: Truncation order of various spatial differencing schemes

3.8. Numerical Examples

In this section results are presented from the optimal finite element Riemann solver for all three angular discretisation schemes implemented. This shows the optimal

methods are general and not specific to a particular angular framework. The performance of both optimal methods is demonstrated in various radiation transport test problems. Figure (3.1) is a plot of the optimal dissipation coefficients derived in section (3.4) as functions of dimensionless total cross-section for both cases of pure diffusion and full Petrov-Galerkin stabilisation. It is seen from this graph that large values of dimensionless cross-section are problematic for the optimal diffusion scheme and thus require a cut-off. This proved necessary for high-order S_N calculations, where very large dimensionless cross-sections were found to arise particularly with the equal weight quadrature set. This is due to the direction cosine set containing smaller values the more ordinates there are on the sphere. Large dimensionless cross-sections could also arise if the length scale of an element is large, although normally this would not be the case. Interestingly, the optimal diffusion scheme was found to be less in need of a cut-off for the P_N and LW_N angular expansion schemes than it was for the S_N scheme. It is seen from figure (3.1) the optimal DPG scheme is much better behaved in the high cross-section limit, and indeed functioned very well without needing any such cut-off. The amount of dissipation beyond which backward advection may occur is also plotted, and this may be used as an additional cut-off to avoid introducing excessive dissipation as discussed in section (3.6.1).

3.8.1. Problem 1: Validation of Optimal Coefficients

A one-dimensional problem was set up for which an analytical solution was available. This was used to validate both expressions for α derived in section (3.4). The problem comprised one element, of length 1 cm and a total cross-section of 10 cm^{-1}. A flux of 1 n cm^{-2} s^{-1} was applied to left boundary to drive the problem. The analytical solution was calculated at a large number of points across the single element to provide direct comparison with the finite element solutions. This comparison is shown in figure (3.2). The standard DG scheme which does not use any weighting generates a negative solution. This is because without stabilisation, a single element does not provide a spatial resolution adequate to resolve the flux gradient. The optimal methods perform significantly better than the standard DG method, both matching the analytical solution exactly at the outlet node. Their gradients suggest that negative solutions would occur at a coarser mesh resolution than they would with the standard DG method.

3.8.2. Problem 2: Heavy Absorber

The first of the two-dimensional cases is problem 2. Figure (3.3) illustrates the 1 cm $\times$ 1 cm solution domain. This problem was driven (a) with a flux of 1 n cm^{-2} s^{-1} along the left boundary of the domain and (b) with a diagonal drive of 1 n cm^{-2} s^{-1} along the left and lower boundaries of the domain. The material was a pure absorber with a cross-section = 5 cm^{-1}. This cross-section is significant, and a reasonable mesh resolution is required to obtain oscillation free solutions. The first set of solutions to this problem were obtained using just one element across the domain, making this problem very demanding on the schemes. Results are shown in figure (3.4a). The gradient across the element is large due to the high absorption and one element is clearly insufficient to resolve it. This causes the standard DG solution to oscillate, producing a negative radiation flux at the outlet nodes. In this first case where there is a parallel drive both optimal methods match the analytical solution at these nodes exactly, avoiding negative solutions. The second case is even more demanding for the schemes, as the lower boundary is driven also and hence there is diagonal advection across the mesh. Results for this are illustrated in figure (3.4b). Here, the optimal schemes do not obtain the analytical solution exactly. However, they almost match it and their gradients across the single element are smaller than the gradient of standard DG. This suggests the optimal methods would remain positive at coarser resolutions than the DG scheme. Figure (3.5) shows angular flux contours for each drive, and was obtained using a mesh that was sufficient to properly resolve the gradient in order to produce well defined contours.

3.8.3. Problem 3: Imposed Source

The solution domain of the second two-dimensional case (problem 3) is illustrated in figure (3.6). This consisted of an isotropically emitting source of 1 n s^{-1} imposed inside a heavy absorber. Both source and absorber used a total cross-section = 5 cm^{-1} throughout the 12 cm $\times$ 12 cm solution domain. Figure (3.7a) compares line-outs of the P_5 neutron scalar flux solution from each scheme through the central source region. A high resolution result that is spatially converged is included for benchmarking each scheme. A 12 element $\times$ 12 element mesh was used for the low resolution calculations, in which the schemes struggle to represent the gradient of the radiation field attenuating from the source. The DG solution oscillates badly near the base of the source peak, and also across the peak itself. The optimal

methods remain well behaved at all points along the line-out. Importantly, they remain positive where the DG solution becomes negative where the flux values are small. In addition, they produce a flatter source peak that is topologically more like the high resolution benchmark. The base of the central source peak is shown in more detail in figure (3.7b), which clearly illustrates the difficulty encountered when stabilisation is not used in this problem. Huge oscillations are seen on the DG solution at the peak which are completely absent from the optimal DPG solution. The optimal diffusion solution shows some evidence of minor oscillation, but obtains a peak value in extremely close agreement with the benchmark. The two optimal methods also appear to work very well when applied to angular expansions other than the P_N. We have optimal DPG/optimal diffusion results for problem 3 for S_6 and also for an LW_1 wavelet. Figures (3.7c) and (3.7d) show the S_6 results. These used an equal-weight quadrature set for the discrete ordinates approximation. Results for the wavelet are shown in figures (3.7e) and (3.7f). These used the linear octahedral modes at order LW_1. The LW_1 approximation provides 24 angular unknowns in 2-D, the same as S_6. This is also consistent with the P_5 calculation which has 21 angular unknowns. Thus each angular method used for problem 3 should be similarly converged in angle. The optimal DPG method behaves very satisfactorily in all the angular frameworks, performing consistently well across the three we have tested it on. Figure (3.8) compares surface plots of the scalar flux solution, for the three angular expansions. Due to the extremely high total cross-section, neutrons emanating from the source are stopped almost immediately. Thus no ray-effects are seen on the S_6 or LW_1 solutions. Figure (3.9) is a contour plot showing the effect of reducing the cross-section in region 1 to 1.0 cm^{-1} and in region 2 to 0.2 cm^{-1}. This problem is sympathetic to the P_N scheme, and P_5 produces extremely smooth solutions. The problem is more demanding for the discrete ordinates method and the wavelet and ray-effects are clearly visible on the solutions from these two methods.

3.8.4. Problem 4: Duct

Figure (3.10) illustrates the solution domain of problem 4. This problem consisted of a 1 cm wide duct of cross-section = 0.5 cm^{-1} through a dense absorber of cross-section = 10 cm^{-1} of dimension 3 cm $\times$ 3 cm. An isotropic source of 1 n s^{-1} was located in the centre of the duct. This problem was solved using a 9 $\times$ 9 element mesh, a resolution poor enough to significantly challenge each method. Scalar

flux line-outs perpendicular to the duct across the centre of the source are shown for the P_3 calculations in figures (3.11a) and (3.11b). A high resolution result is also shown for benchmarking. The standard DG scheme and the optimal diffusion scheme oscillate at the base. The optimal DPG solution does not oscillate even at this very poor resolution and remains positive. The S_6 calculations are shown in figures (3.11c) and (3.11d) and the wavelet in figures (3.11e) and (3.11f). In all three cases, the optimal DPG scheme achieves superior stability. Surface plots of the optimal DPG scalar flux solutions are compared for the three angular schemes in figure (3.12). Rays in the S_6 and LW_1 solutions are attenuated very rapidly by the absorber and are not easily visible. Thus the differences in the scalar flux field between the three angular schemes is unremarkable.

3.9. Conclusions

Two novel discontinuous finite element Riemann methods have been developed for stabilising the stationary form of the first-order Boltzmann transport equation. Both schemes suppress oscillation in demanding two dimensional radiation transport test problems, allowing the capture of sharp gradients in the solution field without this destabilising the differencing scheme. Both are optimal discretisation methods in the sense that they exactly match the analytical solution in one-dimensional problems. They are implemented within a Riemann framework. This framework makes them general so they may be applied to different angular schemes. This has been demonstrated with spherical harmonics, discrete ordinates and a linear octahedral wavelet where the two methods perform consistently well. Error analysis suggests the optimal discontinuous Petrov-Galerkin scheme is second order, the optimal diffusion scheme is first order and the standard discontinuous Galerkin scheme is third order. However, the two optimal discretisation methods reduce spurious oscillations and the tendency to produce negative fluxes when compared to standard discontinuous Galerkin. Numerical evidence suggesting that they can also be more accurate has been presented. The optimal discontinuous Petrov-Galerkin scheme is believed to be superior to the optimal diffusion scheme, since it is more complete mathematically being residual based and more satisfactory physically due to inclusion of the source term. The numerical results shown in this section suggest both optimal schemes form excellent discretisation methods capable of obtaining robust numerical solutions across different radiation regimes. Future interests are developing optimal space-time methods, applying op-

timal dissipation to problems in axisymmetric and three dimensional geometries and parallelisation of the optimal Riemann solver. One notable deficiency of these two linear schemes, however, is that they apply the same length scale in the added term in the equations. It is suggested that some form of linearity be added to the discretisation, for example using the solution gradient to define locally the most appropriate magnitude to use in defining the length scale.

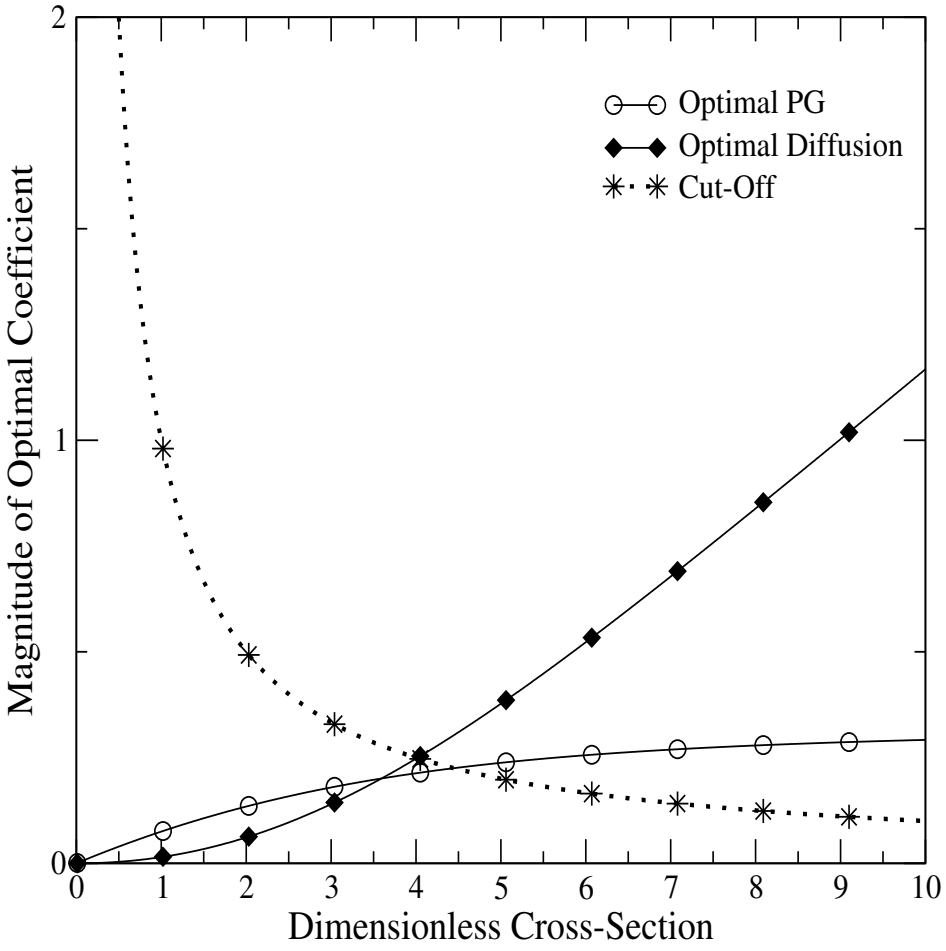


Figure 3.1.: Variation of Optimal Coefficient with Material

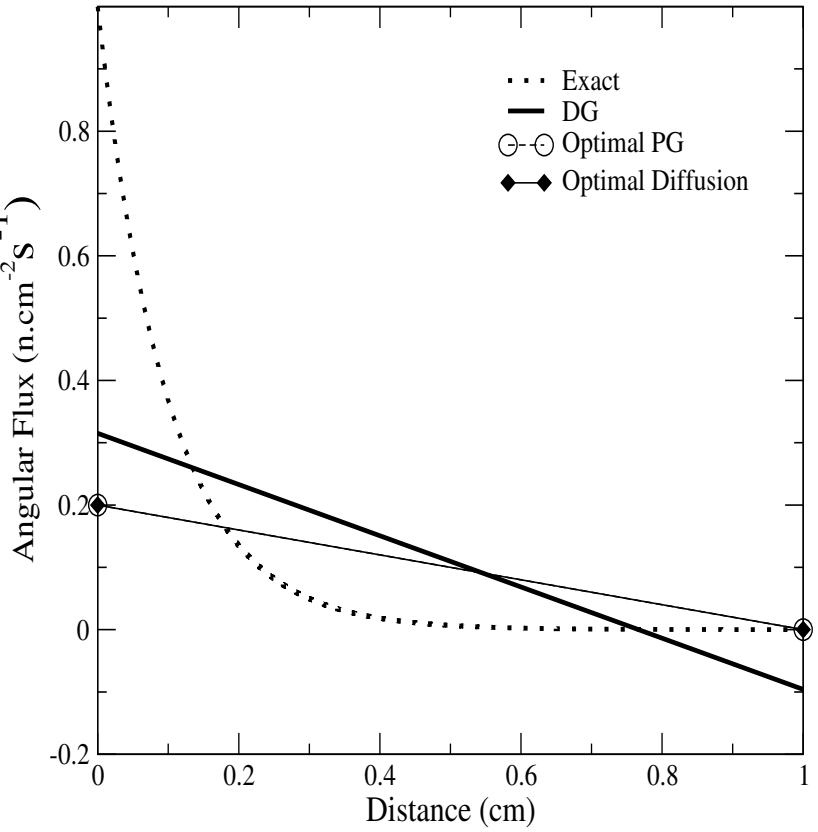
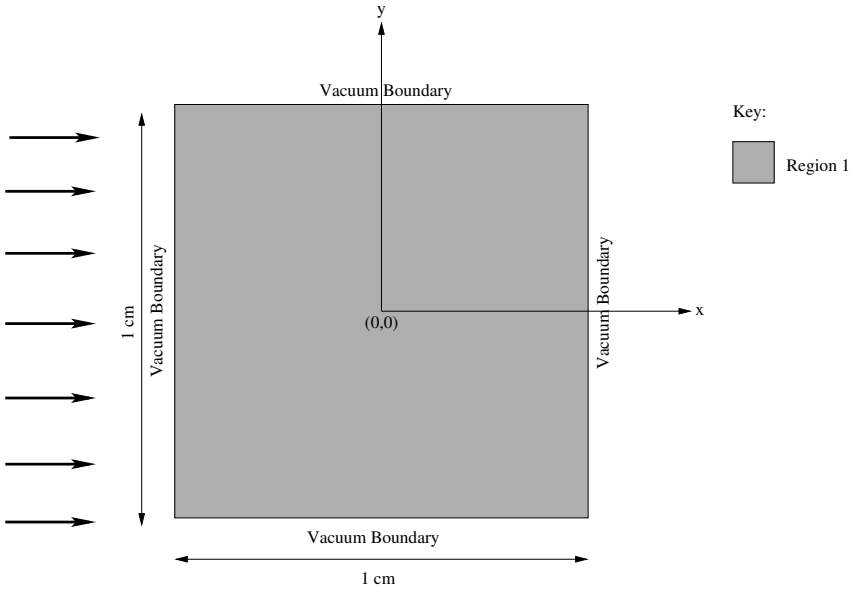
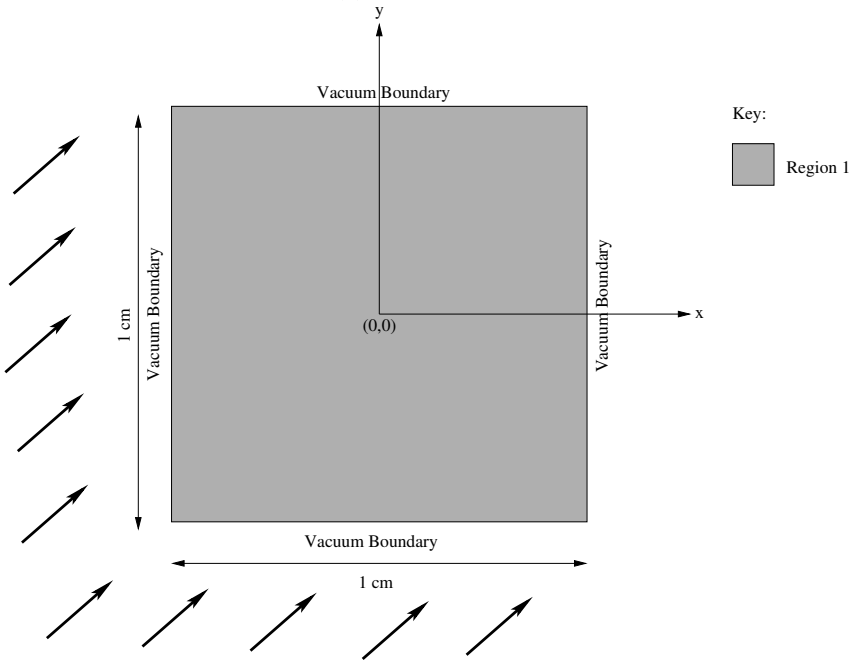


Figure 3.2.: Standard DG solution and the two optimal methods compared with analytical solution using a single element of length 1 cm in one spatial dimension. Use of the optimal coefficients matches the finite element solution with the analytical at the element outlet node. A total cross-section = 10 cm^{-1} was used for this calculation. Note that the optimal PG solution and optimal diffusion solution overlie exactly, and also that the gradient of the DG solution is considerably steeper. This implies the DG solution is more oscillation-prone than either optimal method.

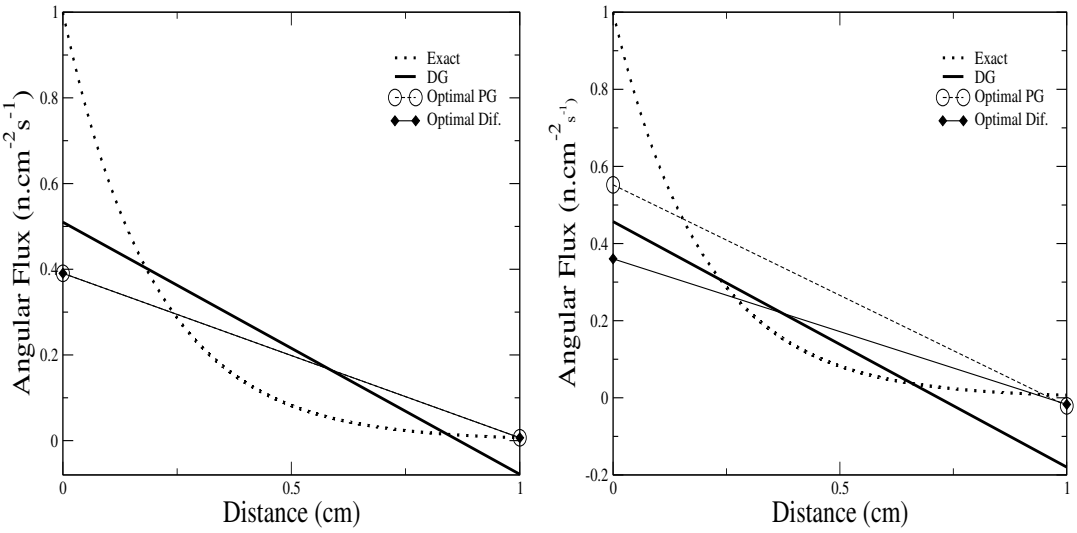


(a) Parallel drive



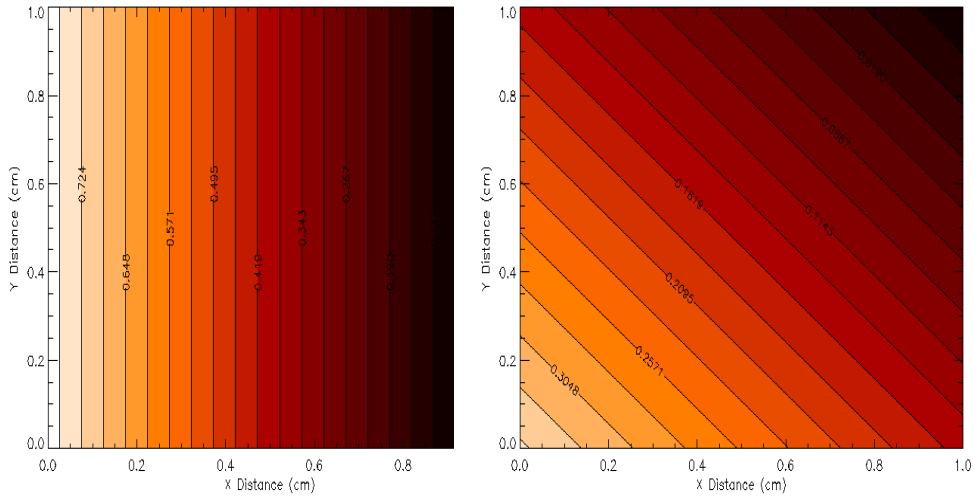
(b) Diagonal drive

Figure 3.3.: The solution domain for problem 2. This problem comprises one region of total cross-section 5 cm^{-1} with a scattering cross-section = 0 cm^{-1} and was driven (a) by a flux of 1 $n{\cdot}cm^{-2}{\cdot}S^{-1}$ along the left hand boundary and (b)by a flux of 1 $n{\cdot}cm^{-2}{\cdot}S^{-1}$ along both the left hand boundary and the lower boundary.



(a) Parallel drive (b) Diagonal drive

Figure 3.4.: Standard and optimal solutions to problem 2 compared with analytical solution for a single element in 2 dimensions of spatial length Δx = Δy = 1 cm and total cross-section = 5 cm^{-1}. Line-outs are along (a) the top edge of problem 2 driven with a parallel beam at x=0 cm, advecting in the x direction only (pseudo 1-D) and (b) the right edge with equal drives at both x = 0 cm and y = 0 cm advecting diagonally across the mesh. Note that in the latter case, the analytical solution is not matched exactly.



(a) Parallel drive (b) Diagonal drive

Figure 3.5.: Scalar flux ($ncm^{-2}s^{-1}$) contour plot of the optimal DG results for (a) the parallel beam problem and (b) for the diagonal beam problem. Contour labels show the magnitude of the scalar flux field at various fronts across the 2-dimensional mesh.

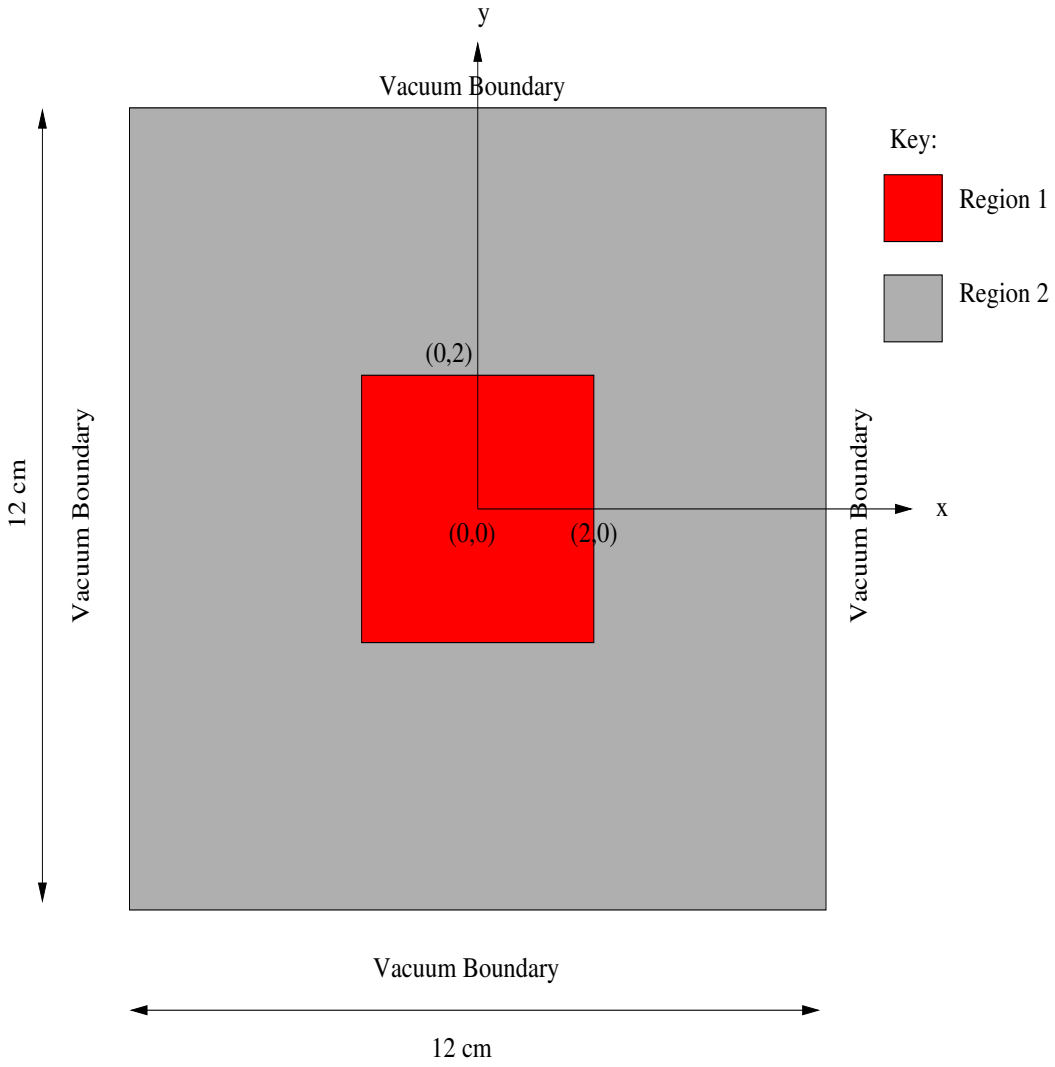
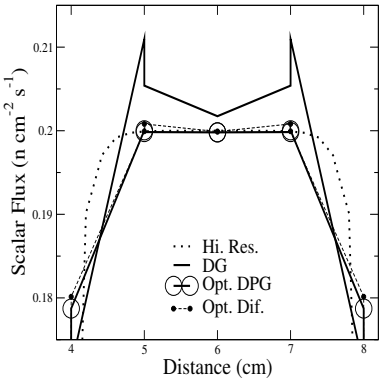
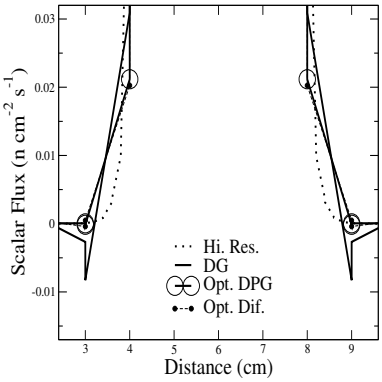


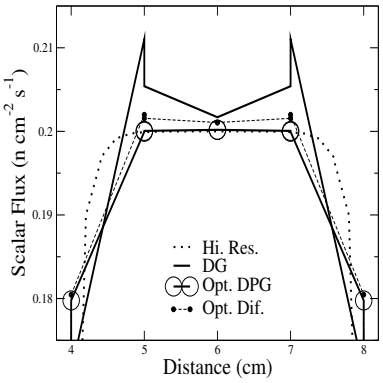
Figure 3.6.: The solution domain for problem 3. This comprised two regions and was calculated on a uniform 12 x 12 element mesh. Region 1 was a neutron source of 1 ns^{-1} and total cross-section = 5 cm^{-1}. Region 2 was not sourced and used a total cross-section = 5 cm^{-1}. A scattering cross-section = 0 cm^{-1} was used for both regions.



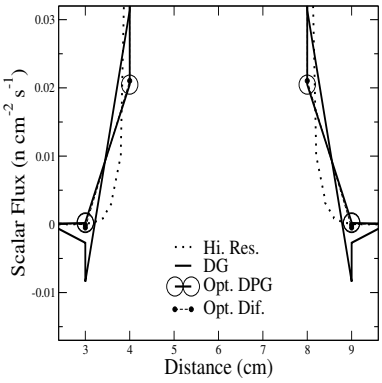
(a) P_5 source peak



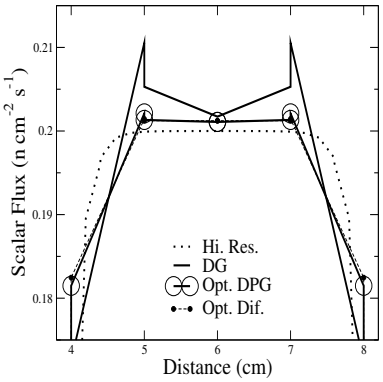
(b) P_5 source base



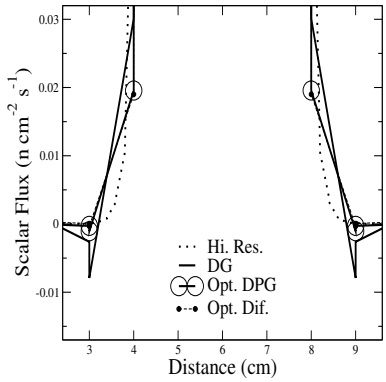
(c) S_6 source peak



(d) S_6 source base

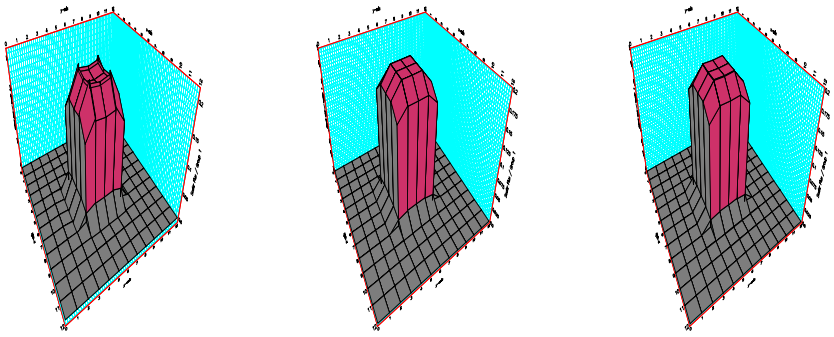


(e) LW_1 source peak

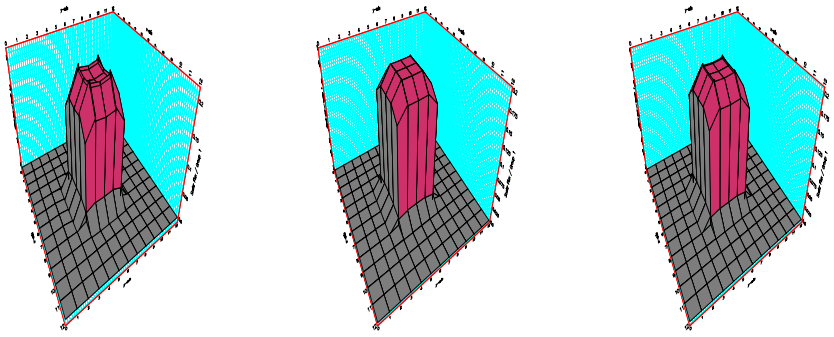


(f) LW_1 source base

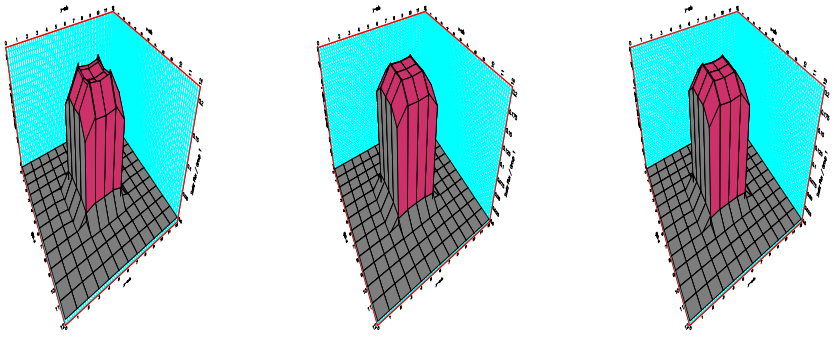
Figure 3.7.: Line-outs of neutron scalar flux across the peak and base of the imposed source region in problem 3 at y=4 cm as a function of x. Standard discontinuous Galerkin and the optimal DG results are compared for P_5 ((a) and (b)) S_6 ((c) and (d)) and LW_1 ((e) and (f)) approximations.



(a) P_5 Standard DG (b) P_5 Optimal PG (c) P_5 Optimal Diffusion

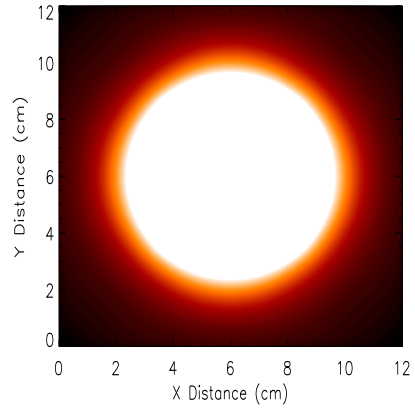


(d) S_6 Standard DG (e) S_6 Optimal PG (f) S_6 Optimal Diffusion

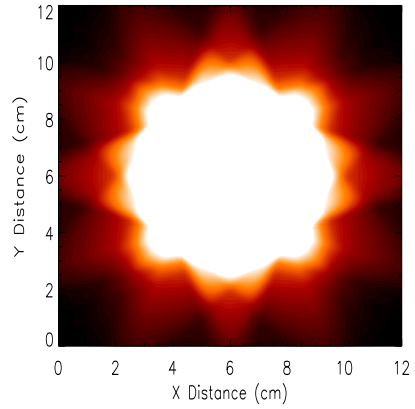


(g) LW_1 Standard DG (h) LW_1 Optimal PG (i) LW_1 Optimal Diffusion

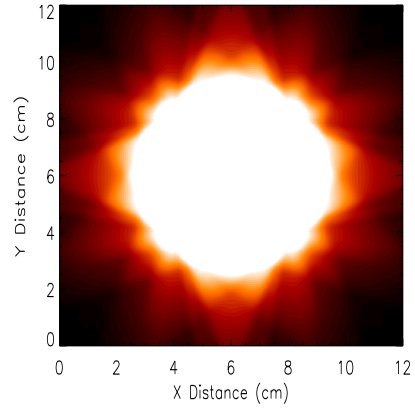
Figure 3.8.: Scalar flux solutions to problem 3 for the different discretisations



(a) P_5 Optimal DPG



(b) S_6 Optimal DPG



(c) LW_1 Optimal DPG

Figure 3.9.: Scalar flux contours of problem 3 for optimal discontinuous Petrov-Galerkin with each angular discretisation scheme when total cross-sections of 1 cm^{-1} and 1/5 cm^{-1} are used in regions 1 and 2 respectively

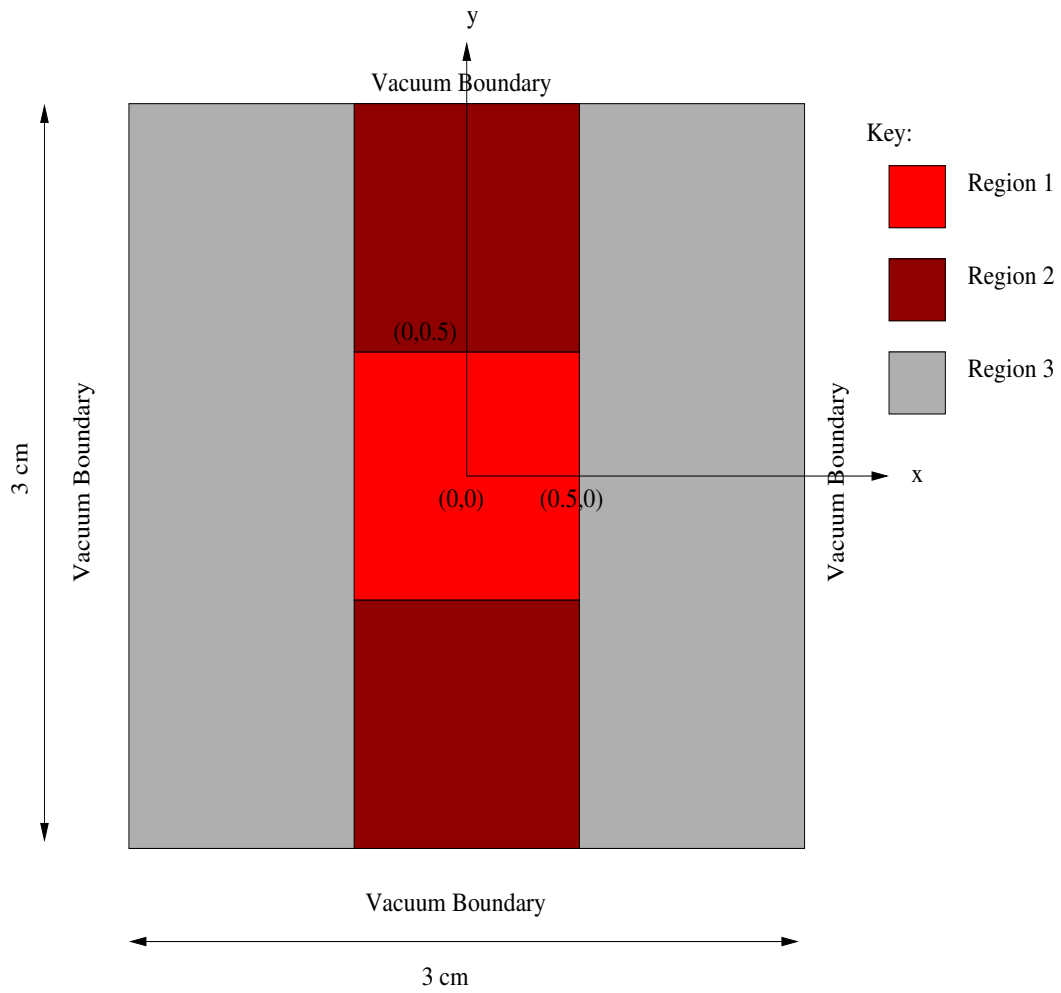
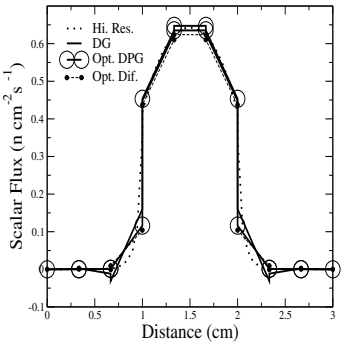
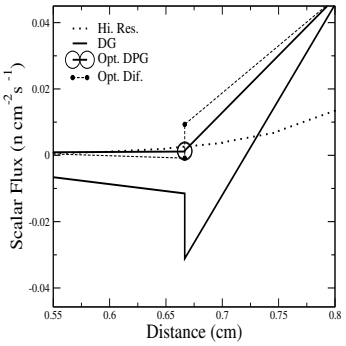


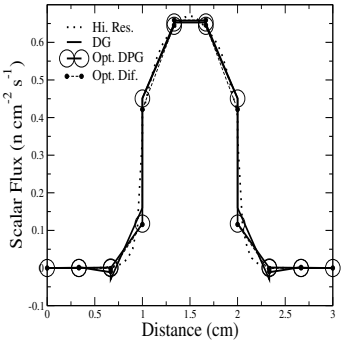
Figure 3.10.: The solution domain of duct problem 4. The channel regions 1 and 2 used a total cross-section of 1/2 cm^{-1} and the absorber region 3 used a total cross-section of 10 cm^{-1}. In region 1 a 1 ns^{-1} isotropic source was imposed. A 9 x 9 element mesh was used to calculate this problem.



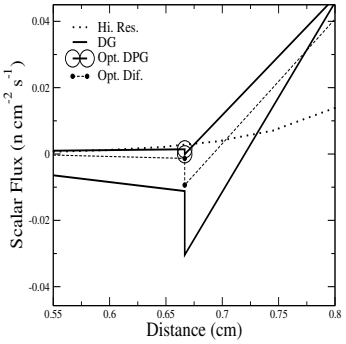
(a) P_5 source peak



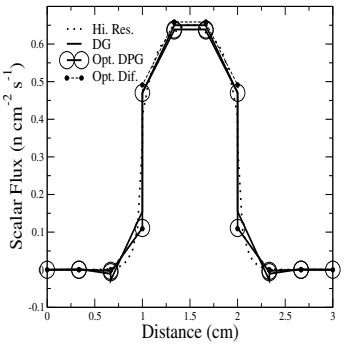
(b) P_5 source base



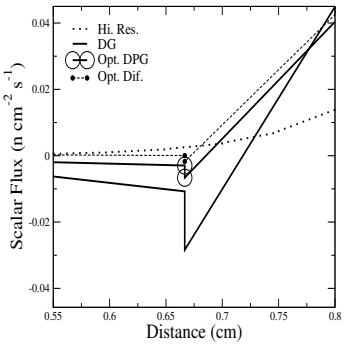
(c) S_6 source peak



(d) S_6 source base

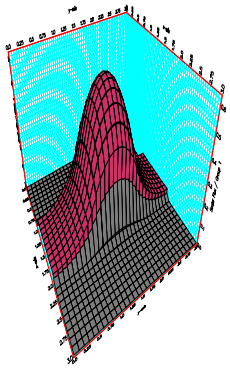


(e) LW_1 source peak

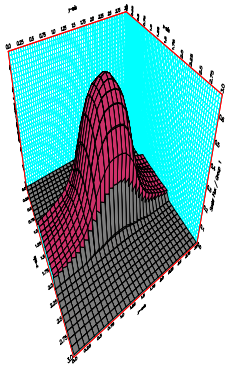


(f) LW_1 source base

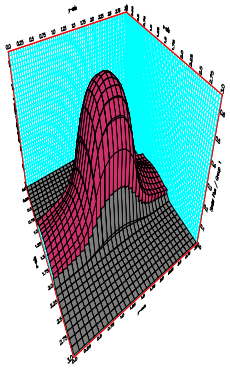
Figure 3.11.: Line-outs of neutron scalar flux across the peak and base of the imposed source region in problem 4 at y=4 cm as a function of x. Standard discontinuous Galerkin and the optimal DG results are compared for P_5 ((a) and (b)) S_6 ((c) and (d)) and LW_1 ((e) and (f)) approximations.



(a) P_5 Optimal DPG



(b) S_6 Optimal DPG



(c) LW_1 Optimal DPG

Figure 3.12.: Optimal DPG scalar flux solutions to problem 4 for the different angular discretisation schemes

CHAPTER 4

NON-LINEAR METHODS

Synopsis

Standard discontinuous Galerkin schemes were demonstrated in the previous chapter to benefit from streamline stabilisation, provided that the amount of artificial dissipation added to the discretisation is limited. This is to prevent more particles being advected backwards than forwards across the element. It is relatively straightforward to avoid this by considering the material properties. However difficulty arises because the amount of dissipation that is optimal can vary from one element to another. For example, in space-time problems if the width of the element is used to limit the amount of dissipation this is good when steady state is reached but may not be suitable in the early stages of a calculation where there is strong transient behaviour. In this case, the element time-step would be a better choice but might result in not enough dissipation being applied once steady-state is reached. Thus it is not always possible with the linear method presented in the previous chapter, to reliably optimise the amount of dissipation added to the system. The present chapter addresses this by adding non-linearity to the optimal methods. This defines locally the most appropriate amount of dissipation to use based on the solution gradient. [1]

[1] This chapter was presented to the M&C 2009 conference, Saratoga Springs, NY, USA May 2008.

Contents

4.1. Introduction

This section describes new non-linear Discontinuous Petrov-Galerkin methods and demonstrates their application to the one-speed Boltzmann Transport Equation (BTE) for space-time problems. The purpose of the methods is to remove unwanted oscillations in the transport solution which occur in the vicinity of sharp flux gradients, while improving computational efficiency and numerical accuracy. This is achieved by applying artificial dissipation in the solution gradient direction, internal to an element using a novel finite element (FE) Riemann approach. The amount of dissipation added acts internal to each element. This is done using a gradient-informed scaling of the advection velocities in the stabilisation term. This makes the method in its most general form is non-linear. In addition, an approach is outlined for applying non-linearity to the sub-grid-scale (SGS) method which relates to the discontinuous Galerkin method used in the present work. The approach is based on the cosine rule between the advection direction in Cartesian space-time and the direction of the solution gradient. The implementation is designed to be independent of angular expansion framework. This is demonstrated for the both discrete ordinates (S_N) and spherical harmonics (P_N) descriptions of the angular variable. Results show the schemes perform consistently well in demanding time-dependent and multi-dimensional radiation transport problems.

The work in the current section combines the Riemann approach with recent optimal finite methods. These add an amount of dissipation such that exact solutions are matched by the finite element solution where they exist (Pain and Goddard, 2000; Merton et al., 2008). These optimal methods are governed by the choice of length scale which characterises the dissipation. The correct choice of length across which to add the dissipation is a crucial issue. On space-time meshes, this has traditionally meant a choice between either the spatial length of an element or the time-step. Use of the spatial length results in the correct amount of numerical dissipation being introduced where the space-time solution has reached steady-state and the temporal modes have diminished. Use of the element time step will result in the correct amount of dissipation being introduced at early time while there is transient behaviour, but assumes the problem is spatially invariant which is usually not the case. Neither choice of length scale is entirely satisfactory, and this has motivated the development of non-linear Petrov-Galerkin schemes. These methods use knowledge of the finite element solution to calculate locally the most appropriate length scale to use. Where the temporal gradients dominate, the time step is selected as the most representative distance. Elsewhere, the width of the element is used. Thus different length scales will be selected in different elements based on local behaviour of the solution. This has motivated development of the scheme in the current section, which uses the solution gradient to define locally the most appropriate dissipation to use. In section (2.2) the time-dependent linear form of the one-speed Boltzmann Transport Equation (BTE) was introduced and discretised using linear discontinuous finite elements, for both the spatial and temporal grids. No assumption is made in this Thesis regarding the angular scheme; a key feature of this work is that it is independent of the choice of basis function describing direction of particle travel. Consequently, the methods described in this chapter operate for arbitrary discretisation in angle. Galerkin weighting is used for both the spatial and temporal projections. A space-time Riemann approach to implementing the boundary conditions is used that is independent of the angular scheme used. The details of this approach were described in section (2.3).

The present chapter has been arranged as follows. In section (4.2) the linear optimal method that was introduced in chapter (3) is briefly reviewed. However, the linear optimal method uses a common choice of length scale throughout the whole problem. This may not be entirely satisfactory as it results in the same type of dissipation being used everywhere throughout the domain. It is this idea that motivates the non-linear methods derived in the present section. Section (4.3) in-

troduces a general non-linear optimal method that is of similar form to the linear optimal method. This method allows the length scales to be defined locally based on local behaviour of the solution. Various other types of non-linear methods are then derived in detail, first for the scalar equations in section (4.4) and then for the general coupled equations in section (4.5). Care needs to be taken when adding dissipation to the discretisation; to avoid dissipating too many particles positivity must be ensured. This is considered in section (4.6). Section (4.7) considers a method of treating voids, as they can be problematic in the stabilisation term. The stabilisation term amounts to additional upwinding of the finite element stencil; this is discussed in section (4.8). An approach to adding the non-linear methods to the Sub-Grid-Scale method, which closely relate to, is considered in section (4.9). Numerical results are presented and discussed in section (4.10). The non-linear scheme is demonstrated on a series of demanding time-dependent and steady-state problems on two dimensional and three dimensional meshes. This is done for discrete ordinates and spherical harmonics on the sphere. Conclusions are drawn in section (4.11).

4.2. Optimal Linear Methods

In many demanding transport applications it is necessary to capture sharp gradients in the solution, for example in the vicinity of material boundaries. The discontinuous Galerkin method allows this to some extent by not enforcing continuity of solution at the element boundaries. Although this is adequate for many types of radiation transport problem, there are cases when it produces unwanted oscillation. It is possible to suppress this behaviour by adding on to the discretisation an upwind term for gradient control and numerical stability. This results in a Petrov-Galerkin (PG) formulation that improves robustness of the transport solution while retaining the accuracy. The general form of the angular discretised equation with this modification may be written:

$$(\mathbf{I} - \mathbf{A} \cdot \nabla \mathbf{P})\mathscr{R} = \mathbf{0}, \tag{4.1}$$

in which $\mathbf{P}$ is an angular stabilisation matrix and $\mathscr{R}$ the residual of the governing equation. The choice of $\mathbf{P}$ determines the type of PG scheme. In general, $\mathbf{P}$ is a function of $\mathbf{A}$ and many choices are to be found in the literature. Popular examples are the full Riemann streamline upwind Petrov-Galerkin (SUPG) method

and even power SUPG method such as that used in the finite element code *RADIANT* (Pain et al., 2006b,a; Sherwin et al., 2006). Dissipation is added internal to an element, and the amount that is added can be optimised in various ways to avoid adding in too much diffusion. This is important, as excessive artificial diffusion can heavily reduce accuracy and convergence of the scheme. One of the earliest optimal methods was published in (Pain and Goddard, 2000) in which the finite element solution matched the exact solution at each outlet node in one spatial dimension. This was achieved by scaling the amount of dissipation added internal to an element with a parameter α calculated from the cross-section. The work was later extended in (Merton et al., 2008) with development of an optimal method for multi-dimensional systems, where an exact solution was not available. This method worked by deriving a one-dimensional system along the streamline direction of the multi-dimensional system. That is to use $A_s = \mathbf{n}^* \cdot \mathbf{A}$ in which $\mathbf{n}^*$ is defined as a face normal would be defined in a control volume method, i.e. $\mathbf{n}^* = \frac{\nabla \mathbf{N}_i}{|\nabla \mathbf{N}_i|}$ where $\mathbf{N}_i$ is the node i basis function. One may use an appropriate Riemann decomposition of $\mathbf{A}_s$. That is to use $A_s = \mathbf{R}_S \Lambda_s \mathbf{L}_S$ where $\mathbf{R}_s$ and $\mathbf{L}_s$ are the left and right eigenvectors of $\mathbf{A}_s$ and Λ_s the diagonal matrix of eigenvalues associated with the Riemann decomposition of $\mathbf{A}_s$. This allows the derivation of a set of one-dimensional waves. One can then write:

$$\Lambda_s \frac{\partial \Psi_s}{\partial s} + \mathbf{H}_s \Psi_s = \mathbf{0}, \tag{4.2}$$

which has analytical solutions, where $\mathbf{H}_s = \mathbf{L}_s \mathbf{H} \mathbf{R}_s$ is the the scattering removal operator mapped into the appropriate Riemann space. One-dimensional optimal methods such as those in (Pain and Goddard, 2000) may then be applied in this space. In (Merton et al., 2008), a stabilisation matrix $\mathbf{P}$ was defined based on these one-dimensional considerations in order to remove oscillations from the multi-dimensional solution. In this method, the artificial dissipation is added internal to an element. This choice of $\mathbf{P}$ has the form:

$$\mathbf{P} = \mathbf{R}_S \Lambda_s^{-1} \mathbf{G}^T \mathbf{R}_G \Lambda_\alpha \mathbf{L}_G \mathbf{G} \Lambda_s^{-1} \mathbf{L}_s, \tag{4.3}$$

in which $\mathbf{G}^T \mathbf{G} = \mathbf{H}_s$ is the Cholesky factorisation of $\mathbf{H}_s$. One forms the matrix $\mathbf{A}_G = \mathbf{G}^{-T} \mathbf{\Lambda}_s \mathbf{G}^{-1}$, using the Cholesky factors of $\mathbf{H}_s$. This has the eigenvalue decomposition $\mathbf{A}_G = \mathbf{R}_G \mathbf{\Lambda}_G \mathbf{L}_G$, where $\mathbf{R}_G$ and $\mathbf{L}_G$ are the matrices containing the right and left eigenvectors respectively, and $\mathbf{\Lambda}_G$ the diagonal matrix of eigen-

values of $\mathbf{A}_G$. The diagonal matrix Λ_α is formed by placing the optimal coefficient for direction i at position i on the main diagonal. That is to use:

$$\mathbf{\Lambda}_{\alpha_{\mu\nu}} = \begin{cases} h|\mathbf{\Lambda}_{G_{\mu\nu}}|\alpha(\hat{\sigma}) & \mu = \nu; \\ \mathbf{0} & \mu \neq \nu, \end{cases} \tag{4.4}$$

where the optimal coefficient is defined as:

$$\alpha = \frac{(6 + 4\hat{\sigma} + \hat{\sigma}^2)e^{-\hat{\sigma}} + 2\hat{\sigma} - 6}{(12 - 6\hat{\sigma}) - (12 + 6\hat{\sigma})e^{-\hat{\sigma}}}. \tag{4.5}$$

Thus the dimensionless cross-section for the set of one-dimensional waves is given by $\hat{\sigma} = \frac{\sigma h}{|\Lambda_G|}$. Inserting the Cholesky factorisation into the angular discretisation ensures that equation (4.2) is diagonal; note that without this approach, the system would not be diagonal for wavelet methods or the discrete ordinates method with anisotropic scattering, due to $\mathbf{H}_s$ being non-diagonal. The Cholesky factorisation of $\mathbf{H}_s$ makes the resulting stabilisation scheme work for arbitrary angular discretisation.

Artificial dissipation in PG methods is characterised by the element length scale which determines the distance across which the dissipation acts. In (Pain and Goddard, 2000) and (Merton et al., 2008) the length of the element in space was used. However, this choice becomes in appropriate where there is transient behaviour in which case the element time-step could be a better choice. Where both spatial and temporal gradients are important, neither the spatial length of the element nor the time-step are entirely satisfactory. Thus there is ambiguity regarding the type of dissipation to use. Section (4) introduces non-linearity into the method of (Merton et al., 2008) to define a more appropriate type of dissipation.

4.3. Optimal Non-Linear Methods

Conventional non-linear SUPG methods, such as that in the finite element code RADIANT, use an amount of dissipation that is made optimal in the gradient direction, but acts in the streamline direction (Eaton, 2004). This is capable of shock capture, provided the gradient and streamline are not perpendicular. In the non-linear method presented in the current section, the added dissipation acts in the direction of the solution gradient. This approach allows shock capture even if the gradient and streamline are perpendicular. The dissipation used is made optimal

with a method based on that of (Merton et al., 2008). In section (4.2), an optimal stabilisation matrix $\mathbf{P}$ was introduced based on the optimal coefficient α in the Riemann space of $\mathbf{A}_s = \mathbf{R}_s\Lambda_s\mathbf{L}_s$. This stabilisation matrix describes the magnitude of numerical dissipation that is introduced to an element to stabilise the solution. The coefficient optimises the magnitude of the dissipation for a particular choice of length scale. In the linear DPG method described in section (4.2), the spatial length of the element is used for this length scale. A disadvantage of this approach is that the same choice of length scale is used everywhere within the solution domain. Use of the element spatial width will become optimal as transient solutions reach steady-state but may be inappropriate at early time while there is time-dependent behaviour. Using the element time-step as the length scale will produce optimal dissipation where the temporal gradients dominate over the spatial gradients but becomes inappropriate once steady-state is reached. In general, the direction of the solution gradient $\mathbf{n} \cdot \frac{\nabla\Psi}{|\nabla\Psi|}$ will vary across the problem and may change with time also. Linear methods such as that discussed in section (4.2) have no knowledge of the solution gradient, and consequently are not guaranteed to be optimal at all points across the domain at all times. Another deficiency they have, also arising from the fact they have no local knowledge of solution gradient, is they are not capable of shock-capture; they introduce dissipation only along the direction of the $\nabla\mathbf{N}_i$ term in the stabilisation. To enable shock-capture, local gradients of the solution must be taken into consideration making the scheme non-linear. The direction in which the dissipation is optimal is then defined locally depending on the solution gradient. In the present section, a method is formulated that is optimal in a direction defined locally depending on the direction of the solution gradient in space-time. Equation (4.1) may be generalised to the form:

$$(\mathbf{I} - \mathbf{A}^* \cdot \nabla\mathbf{P}^*)\mathscr{R} = \mathbf{0}, \tag{4.6}$$

in which $\mathbf{A}^*$ is a modified advection velocity defined in equation (4.39). The stabilisation matrix $\mathbf{P}^*$ may be defined in an expression that is of the same form as that used to define $\mathbf{P}$ used in the linear optimal method, and it may be written:

$$\mathbf{P}^* = \mathbf{R}^*_S\Lambda^{*-1}_s\mathbf{G}^T\mathbf{R}^*_G\Lambda_\alpha\mathbf{L}^*_G\mathbf{G}\Lambda^{*-1}_s\mathbf{L}^*_s, \tag{4.7}$$

in which $\mathbf{R}^*_s$ and $\mathbf{L}^*_s$ are the matrices containing the right and left eigenvectors of $\mathbf{A}^*_s = \frac{\nabla\mathbf{N}_i}{|\nabla\mathbf{N}_i|}\cdot\mathbf{A}^*$ respectively, and Λ^*_s the diagonal matrix of eigenvalues associated

with the Riemann decomposition of $\mathbf{A}_s^*$. Thus $\mathbf{P}^*$ is the same function of $\mathbf{A}^*$ that $\mathbf{P}$ is of $\mathbf{A}$ in equation (4.1). Therefore, the underlying optimal formulation is identical in the optimal non-linear PG method and the optimal linear method. Equation (4.6) may be used as a general form for SUPG equations; note that using $\mathbf{A}^* = \mathbf{A}$ and $\mathbf{P}^* = \mathbf{P}$ results in the standard definition. In the current work, the above definition used for $\mathbf{A}^*$ is akin to that used by Donea and co-workers (Donea and Huerta, 2003). That is to project the advection velocity $\mathbf{A}$ onto the gradient of the solution. This enables shock capture by ensuring the diffusion acts along the direction of the solution gradient which varies locally rather than acting along the direction of the streamline at all points in the domain. One projects the advection velocity in this fashion by considering the two vectors $\mathbf{A} = (\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z)^T$ and $\nabla\Psi = (\frac{\partial\Psi}{\partial x}, \frac{\partial\Psi}{\partial y}, \frac{\partial\Psi}{\partial z})^T$, and applying the cosine rule. Note that each component of $\nabla\Psi$ is itself a vector of length $\mathcal{M}$, component m of which refers to moment m in the angular expansion of Ψ. Each component of $\mathbf{A}$ is an $\mathcal{M} \times \mathcal{M}$ matrix. An alternative formulation of the optimal non-linear method, that provides a simpler way of embedding the optimal coefficient into the non-linear term is discussed in section (4.4) for the ray-like scalar equation and in section (4.5) for the more general coupled moment equations. In each section, a diffusion form of the non-linear stabilisation is also derived.

4.4. Scalar Equations

The time-dependent transport equation may be written in the form of a scalar equation, that is to consider a single moment. This applies to a single ray streaming through the problem and provides an easy framework to formulate non-linear methods before considering the more general coupled form in section (4.5), which applies to any angular discretisation scheme such as P_N or LW_N as well as to S_N. The scalar time-dependent equation is written as:

$$\mathbf{a}_{xt} \cdot \nabla_{xt}\psi + \sigma\psi = s. \tag{4.8}$$

Thus, in one-dimensional geometry $\mathbf{a}_{xt} = (a_t, a_x)^T$ and $\nabla_{xt}\psi = (\frac{\partial\psi}{\partial t}, \frac{\partial\psi}{\partial x})^T$. In three dimensional geometry these vectors become $\mathbf{a}_{xt} = (a_t, a_x, a_y, a_z)^T$ and $\nabla\psi = (\frac{\partial\psi}{\partial t}, \frac{\partial\psi}{\partial x}, \frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial z})^T$. The one-dimensional partial differential equation with

time-dependence included becomes:

$$a_t \frac{\partial \psi}{\partial t} + a_x \frac{\partial \psi}{\partial x} + \sigma \psi = s. \tag{4.9}$$

The cosine rule may be applied to the two vectors $\mathbf{a}_{xt} = (a_t, a_x)^T$ and $\nabla_{xt}\psi = (\frac{\partial \psi}{\partial t}, \frac{\partial \psi}{\partial x})^T$, in which θ_a is the angle subtended between them. This allows the projection of the vector $\mathbf{a}_{xt}$ onto the vector $\nabla_{xt}\psi$ to be derived. The angle θ_a is illustrated in figure (4.1), for the one-dimensional time-dependent case. Applying

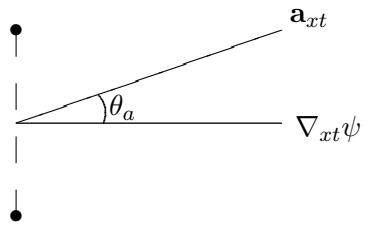


Figure 4.1.: Using the cosine rule between the vectors $\mathbf{a}_{xt}$ and $\nabla_{xt}\psi$

the cosine rule to $\mathbf{a}_{xt}$ and $\nabla_{xt}\psi$, one may state that $\cos(\theta_a) = \frac{\mathbf{a}_{xt} \cdot \nabla_{xt}\psi}{|\mathbf{a}_{xt}||\nabla_{xt}\psi|}$. The quantities $|\mathbf{a}_{xt}|$ and $|\nabla_{xt}\psi|$ are the Euclidean norms of the two vectors, respectively. This type of norm is simply the geometric length (or magnitude) of each vector. Note, therefore, that $|\mathbf{a}_{xt}| = \sqrt{a_t^2 + a_x^2}$ and $|\nabla_{xt}\psi| = \sqrt{\frac{\partial \psi}{\partial t}^2 + \frac{\partial \psi}{\partial x}^2}$. The projection of $\mathbf{a}_{xt}$ on $\nabla_{xt}\psi$ may be written $\mathbf{a}^*_{xt} = |\mathbf{a}_{xt}|\mathbf{n}\cos(\theta_a)$, where $\mathbf{n}$ in this definition is the unit normal in the direction of $\nabla_{xt}\psi$, i.e. $\mathbf{n} = \frac{\nabla_{xt}\psi}{|\nabla_{xt}\psi|}$. One readily obtains the expression:

$$\mathbf{a}^*_{xt} = \frac{(\mathbf{a}_{xt} \cdot \nabla_{xt}\psi)\nabla_{xt}\psi}{||\nabla_{xt}\psi||^2}. \tag{4.10}$$

From equation (4.10), one can see that:

$$\mathbf{a}^*_{xt} \cdot \nabla_{xt}\psi = \mathbf{a}_{xt} \cdot \nabla_{xt}\psi, \tag{4.11}$$

or:

$$\left(\frac{(\mathbf{a}_{xt} \cdot \nabla_{xt}\psi)\nabla_{xt}\psi}{||\nabla_{xt}\psi||^2} \right) \cdot \nabla_{xt}\psi = \mathbf{a}_{xt} \cdot \nabla_{xt}\psi, \tag{4.12}$$

where $||\nabla_{xt}\psi||^2 = (\frac{\partial\psi}{\partial t}^2 + \frac{\partial\psi}{\partial x}^2)$ is the square of the 2-norm of $\nabla_{xt}\psi$. The optimal non-linear presented in the section (4.3) may suffer, in certain classes of problem, from slow convergence of the iteration on the stabilisation term. In general this can be mitigated by applying a relaxation technique. A non-linear Petrov-Galerkin (NLPG) approach is outlined here that does not use the optimal coefficient. This might offer a more rapid convergence in these types of problems. An alternative definition of the stabilisation parameter p_{xt} is used that instead respects the length scale of the element. As with the optimal non-linear method, this scheme defines locally the an appropriate numerical dissipation to include in the discretisation. The non-linear Petrov-Galerkin approach uses a modified form of the differential equation, obtained by adding an upwind non-linear term to the discretisation. This equation has the form:

$$(\mathbf{I} - \nabla_{xt} \cdot \mathbf{a}^*_{xt} p^*_{xt})(\mathbf{a}_{xt} \cdot \nabla_{xt}\psi + \sigma\psi - s) = 0. \tag{4.13}$$

Testing equation (4.13) with a space-time basis function N_{xti} and integrating over a single element V_E with boundary Γ_E and applying integration by parts results in:

$$\int_{V_E} dV_{xt} N_{xti} r - \oint_{\Gamma_E} d\Gamma_{xt} N_{xti} \left(\mathbf{n}_{xt} \cdot \mathbf{a}_{xt}\right)^- (\psi - \psi_{bc})$$
$$+ \int_{V_E} dV_{xt} (\nabla_{xt} N_{xti}) \cdot \mathbf{a}^*_{xt} p^*_{xt} r + \oint_{\Gamma_E} d\Gamma_{xt} N_{xti} \mathbf{n}_{xt} \cdot \mathbf{a}^*_{xt} p^*_{xt} r = 0, \tag{4.14}$$

with a finite element expansion $\psi = \sum_{j=1}^{\mathcal{N}} N_{xtj}\psi_j$ and $r = \mathbf{a} \cdot \nabla_{xt}\psi + \sigma\psi - s$ and $\mathbf{n}_{xt}$ is the normal to the element in space-time and $(\mathbf{n}_{xt} \cdot \mathbf{a}_{xt})^- = min\{0, \mathbf{n}_{xt} \cdot \mathbf{a}_{xt}\}$ enables the incoming boundary information to be defined. It is convenient to apply a zero boundary condition to the residual $r = 0$, on Γ_E, which is a natural solution to the residual equation and results in:

$$\int_{V_E} dV_{xt} N_{xti} r - \oint_{\Gamma_E} d\Gamma_{xt} (\mathbf{n}_{xt} \cdot \mathbf{a}_{xt})^- N_{xti} (\psi - \psi_{bc})$$
$$+ \int_{V_E} dV_{xt} (\nabla_{xt} N_{xti}) \cdot \mathbf{a}^*_{xt} p^*_{xt} r = 0. \tag{4.15}$$

The scalar p^*_{xt} is a function of $\mathbf{a}^*_{xt}$ and respects the size and shape of the elements. There are a number of definitions for this that can be used that are quite satisfactory, for example:

$$p^*_{xt} = \frac{1}{4}(|\mathbf{a}^*_{xt} \cdot \nabla_{xt} N_{xti}|)^{-1}. \tag{4.16}$$

This expression is obtained from the Riemann finite element method, (Pain et al., 2006b). This builds in to the non-linear term a preference for the shape function which is aligned with the direction of $\mathbf{a}^*_{xt}$, for elements with equal sized faces. Alternatively one can produce an p^*_{xt} that is independent of i with:

$$p^*_{xt} = min_k\{\frac{1}{4}(|\mathbf{a}^*_{xt} \cdot \nabla_{xt} N_{xtk}|)^{-1}\}. \tag{4.17}$$

One will note that this expression uses the length scale of the element in the direction of $\mathbf{a}^*_{xt}$. Using the 2-norm and the finite element space-time Jacobian matrix $\mathbf{J}_{xt}$ an alternative expression to (4.17) can be found:

$$p^*_{xt} = \frac{1}{4}(||\mathbf{J}^{-1}_{xt}\mathbf{a}^*_{xt}||_2)^{-1}. \tag{4.18}$$

The value of p^*_{xt} can be adjusted to ensure that the resulting value of p^*_{xt} is not so large that one advects more backwards than forwards in the resulting discrete system of equations using:

$$p^*_{xt} = min\{\frac{1}{\sigma + \epsilon}, \frac{1}{4}(||\mathbf{J}^{-1}_{xt}\mathbf{a}^*_{xt}||_2)^{-1}\}, \tag{4.19}$$

in which $\sigma + \epsilon > 0$ and ϵ is a small positive number that ensures one does not divide by zero $\sigma = 0$ e.g. $\epsilon = 1 \times 10^{-10}$. This handles voids in a robust and simple manner. Continuous Petrov-Galerkin formulations use a factor of $\frac{1}{2}$ instead of $\frac{1}{4}$ in the above. This correctly centres the equation residual at the centre of mass of the basis function, for continuous finite element representations. In the present work, where discontinuous finite elements are used to formulate the space-time discretisation, the centre of mass of the basis function is centred at a distance of $\frac{\Delta x}{4}$ from the upwind boundary of the element. This discussed in detail in section (4.8). In the traditional Petrov Galerkin method $\mathbf{a}^*_{xt} = \mathbf{a}_{xt}$ in the above and p_{xt} replaces p^*_{xt}.

4.4.1. Diffusion Form

Rather than using the NLPG scheme which is residual-based, one can use a non-linear diffusion term in the discretisation. This may give very satisfactory results in problems that have strong diffusive modes, and control oscillation by damping. It may offer a more rapid convergence of the non-linear term due to the ability it has to suppress unphysical behaviour. The residual based NLPG method is likely to be more sensitive to shocks in the solution which could slow convergence of the non-linear term. Therefore, a diffusion form of the non-linear method has been formulated. This may be written:

$$\int_{V_E} dV_{xt} N_{xti} r - \oint_{\Gamma_E} d\Gamma_{xt} N_{xti} (\mathbf{n}_{xt} \cdot \mathbf{a}_{xt})^{-} (\psi - \psi_{bc}) + \int_{V_E} dV_{xt} (\nabla_{xt} N_{xti})^T \nu \nabla_{xt} \psi = 0, \tag{4.20}$$

in which the scalar diffusion coefficient is:

$$\nu = \frac{(\mathbf{a}_{xt} \cdot \nabla_{xt} \psi) p^*_{xt} r}{||\nabla_{xt} \psi||^2}. \tag{4.21}$$

The diffusion ν can be amended to ensure positive diffusion with:

$$\nu = \frac{max\{0, (\mathbf{a}_{xt} \cdot \nabla_{xt} \psi) p^*_{xt} r\}}{||\nabla_{xt} \psi||^2}, \tag{4.22}$$

or working only with the residual by replacing $\mathbf{a}_{xt} \cdot \nabla_{xt} \psi$ with the residual r which results in:

$$\nu = \frac{r p^*_{xt} r}{||\nabla_{xt} \psi||^2}. \tag{4.23}$$

The diffusion coefficient ν is always non-negative because p^*_{xt} is non-negative. Equation (4.23) for the diffusivity can be derived by re-defining $\mathbf{a}^*_{xt}$ in equation (4.10) to:

$$\mathbf{a}^*_{xt} = \frac{r \nabla_{xt} \psi}{||\nabla_{xt} \psi||^2}. \tag{4.24}$$

One will notice therefore that $\mathbf{a}^*_{xt} \cdot \nabla_{xt} \psi = \frac{r \nabla_{xt} \psi}{||\nabla_{xt} \psi||^2} \cdot \nabla_{xt} \psi = r$.

4.4.2. Time-Dependence

Unphysical oscillation may occur in the time direction, as well as in the spatial dimensions. Non-linearity may therefore be required in the temporal discretisation in addition to the spatial. To formulate the non-linear method in space-time, one may assume the time variation is discretised using the two level θ-method. In this case, the residual r may be written:

$$r = a_t \frac{\psi^{n+1} - \psi^n}{\Delta t} + \mathbf{a} \cdot \nabla \psi^{n+\theta} + \sigma \psi^{n+\theta} - s^{n+\theta}, \tag{4.25}$$

with $\mathbf{a} = (a_x \;\; a_y \;\; a_z)^T$ and $\psi^{n+\theta} = \theta \psi^{n+1} + (1-\theta)\psi^n$ and also defining:

$$\nabla_{xt} \psi = (\frac{\psi^{n+1} - \psi^n}{\Delta t}, \; (\nabla \psi^{n+\theta})^T)^T. \tag{4.26}$$

Using this definition (equation (4.26)) enables the mechanics of space-time discretisation to be applied, for example:

$$\mathbf{a}^*_{xt} = (a^*_t, \; \mathbf{a}^{*T})^T = \frac{(\mathbf{a}_{xt} \cdot \nabla_{xt} \psi) \nabla_{xt} \psi}{||\nabla_{xt} \psi||_2^2}, \tag{4.27}$$

and:

$$p^*_{xt} = min\{\frac{1}{\sigma + \epsilon}, \frac{1}{4}(||\mathbf{J}^{-1} \mathbf{a}^*||_2)^{-1}\}, \tag{4.28}$$

in which $\mathbf{J}$ is the block part of the matrix $\mathbf{J}_{xt}$ that is associated with Cartesian space. The stabilised discrete equations in diffusion form can be expressed by only applying diffusion in Cartesian space:

$$\int_{V_E} dV \left(N_i r + \nabla N_i^T \nu \nabla \psi^{n+1} \right) - \oint_{\Gamma_E} d\Gamma N_i (\mathbf{n} \cdot \mathbf{a})^- (\psi^{n+\theta} - \psi^{n+\theta}_{bc}) = 0, \tag{4.29}$$

or in a form where one might apply integration by parts of the transport terms once:

$$\begin{aligned} \int_{V_E} dV N_i \left(a_t (\frac{\psi^{n+1} - \psi^n}{\Delta t}) + \sigma \psi^{n+\theta} - s^{n+\theta} \right) - \int_{V_E} dV \nabla \cdot (N_i \mathbf{a}) \psi^{n+\theta} \\ + \oint_{\Gamma_E} d\Gamma N_i (\mathbf{n} \cdot \mathbf{a})^- (\psi^{n+\theta}_{bc}) + \oint_{\Gamma_E} d\Gamma N_i (\mathbf{n} \cdot \mathbf{a})^+ \psi^{n+\theta} \\ + \int_{V_E} dV (\nabla N_i)^T \nu \nabla \psi^{n+1} = 0. \end{aligned} \tag{4.30}$$

4.4.3. Optimal Petrov-Galerkin Methods

In one-dimension, suppose the transport equation is written:

$$a_x \frac{\partial \psi}{\partial x} + \sigma \psi = s. \tag{4.31}$$

Then defining the non-dimensional absorption coefficient, in which Δx is the size of the finite elements:

$$\hat{\sigma} = \frac{\Delta x \sigma}{|a_x|}. \tag{4.32}$$

The optimal p^* which matches the analytical solution at the outlet DG nodes as demonstrated in chapter (3) (for linear DG elements) is written:

$$p^* = \alpha(\hat{\sigma}) \frac{\Delta x}{|a_x|}, \tag{4.33}$$

with the optimal coefficient as defined in section (3.4.1):

$$\alpha(\hat{\sigma}) = min\{\frac{1}{\hat{\sigma}}, \ \frac{(6 + 4\hat{\sigma} + \hat{\sigma}^2) + (2\hat{\sigma} - 6)e^{\hat{\sigma}}}{(12 - 6\hat{\sigma})e^{\hat{\sigma}} - (12 + 6\hat{\sigma})}\}. \tag{4.34}$$

In the above equation the value of $\alpha(\hat{\sigma})$ has been limited to ensure that the resulting value of p^* is not so large that we have more transport backwards than forwards in the resulting discrete system of equations. Note that the optimal coefficient was defined based on one-dimensional considerations (section (3.4.1)). For multi-dimensional problems the non-dimensional absorption coefficient is written:

$$\hat{\sigma} = (||\mathbf{J}_{xt}^{-1} \mathbf{a}_{xt}^*||_2)^{-1} \ \sigma, \tag{4.35}$$

and a multi-dimensional form for the stabilisation matrix obtained:

$$p_{xt}^* = \alpha(\hat{\sigma}) \ (||\mathbf{J}_{xt}^{-1} \mathbf{a}_{xt}^*||_2)^{-1}. \tag{4.36}$$

4.5. Coupled Equations

In this section, the formulation is generalised to coupled equations that contain $\mathcal{M}$ angular moments. This applies to arbitrary basis functions in the direction of particle travel, rather than to a single ray equation considered previously in section (4.4). The coupled time-dependent equations may be written in their angular

discrete form:

$$\mathbf{A}_{xt} \cdot \nabla_{xt}\Psi + \mathbf{H}\Psi = \mathbf{s}, \tag{4.37}$$

in which $\mathbf{H}$ is a positive semi-definite and diagonal scattering-removal matrix operator. Thus in one-dimension $\mathbf{A}_{xt} = (\mathbf{A}_t \quad \mathbf{A}_x)^T$ and in three dimensions $\mathbf{A}_{xt} = (\mathbf{A}_t \quad \mathbf{A}_x \quad \mathbf{A}_y \quad \mathbf{A}_z)^T$. The one-dimensional equation that includes time-dependence may be written:

$$\mathbf{A}_t \frac{\partial \Psi}{\partial t} + \mathbf{A}_x \frac{\partial \Psi}{\partial x} + \mathbf{H}\Psi = \mathbf{s}. \tag{4.38}$$

The cosine rule may be applied to the two vectors $\mathbf{A}_{xt}$ and $\nabla_{xt}\psi$ to obtain the projection of $\mathbf{A}_{xt}$ onto $\nabla_{xt}\psi$, in exactly the same manner as was done for the scalar equations in section (4.4). For coupled equations, this projection may be written:

$$\mathbf{A}^*_{xt} = \mathbf{V}(\mathbf{A}_{xt} \cdot \nabla_{xt}\Psi)\mathbf{V}(||\nabla_{xt}\Psi||_2^2)^{-1}\nabla_{xt}\Psi. \tag{4.39}$$

Thus:

$$\mathbf{A}^*_{xt} \cdot \nabla_{xt}\Psi = \mathbf{A}_{xt} \cdot \nabla_{xt}\Psi, \tag{4.40}$$

or:

$$\left(\mathbf{V}(\mathbf{A}_{xt} \cdot \nabla_{xt}\Psi)\mathbf{V}(||\nabla_{xt}\Psi||_2^2)^{-1}\nabla_{xt}\Psi\right) \cdot \nabla_{xt}\Psi = \mathbf{A}_{xt} \cdot \nabla_{xt}\Psi, \tag{4.41}$$

in which $\mathbf{V}(\mathbf{g})$ is a diagonal matrix in which $\mathbf{V}(\mathbf{g})_{\mu\mu} = \mathbf{g}_\mu$ and the vector $||\nabla_{xt}\Psi||_2^2$ is such that the μ^{th} entry is $||\nabla_{xt}\Psi||^2_{2\mu} = (\nabla_{xt}\Psi_\mu) \cdot (\nabla_{xt}\Psi_\mu)$. Note that in this notation, $\mathbf{g}$ is a vector that contains the argument passed in to the operator $\mathbf{V}$. For example, $\mathbf{g} = \mathbf{A}_{xt} \cdot \nabla_{xt}\Psi$ or $\mathbf{g} = ||\nabla_{xt}\Psi||_2^2$. Since the matrix $\mathbf{A}^*_{xt}$ has a block diagonal structure the transport equations $\mathbf{A}^*_{xt} \cdot \nabla_{xt}\Psi + \mathbf{H}\Psi - \mathbf{s} = 0$ are a set of $\mathcal{M}$ independent scalar equations in which the μ^{th} scalar equation is:

$$a^*_{t\,\mu}\frac{\partial \Psi_\mu}{\partial t} + a^*_{x\mu}\frac{\partial \Psi_\mu}{\partial x} + a^*_{y_\mu}\frac{\partial \Psi_\mu}{\partial y} + a^*_{z\mu}\frac{\partial \Psi_\mu}{\partial z} + \sigma_\mu\Psi_\mu = \mathbf{s}_\mu, \tag{4.42}$$

and $a^*_{t\,\mu} = \mathbf{A}^*_{t\,\mu\mu}$, $a^*_{x\mu} = \mathbf{A}^*_{x\mu\mu}$, $a^*_{y_\mu} = \mathbf{A}^*_{y_{\mu\mu}}$, $a^*_{z\mu} = \mathbf{A}^*_{z\mu\mu}$, $\sigma_\mu = \mathbf{H}_{\mu\mu}$ and Ψ_μ is the μ^{th} scalar in the vector Ψ. In equation (4.42) it is assumed that $\mathbf{H}$ is a diagonal. If it is not then the system of equations can be easily manipulated so as to diagonalise $\mathbf{H}$, see section (4.5.3). Since the equations have been uncoupled then the scalar equation methods described in the previous section can now be applied. Note that in the case of coupled equations, $||\nabla_{xt}\psi||^2$ becomes a vector of length

$\mathcal{M}$, $(\mathbf{A}_{xt} \cdot \nabla_{xt}\psi)$ a vector of length $\mathcal{M}$ and $\mathbf{A}^*_{xt}$ an $\mathcal{M} \times \mathcal{M}$ diagonal matrix. The definition of $\mathbf{A}^*_{xt}$ may be written:

$$\mathbf{A}^*_{\mu,\nu} = \begin{cases} \frac{(\mathbf{A}\cdot\nabla\Psi)_\mu \nabla\Psi_\mu}{||\nabla\Psi||^2_\mu} & \mu = \nu; \\ 0 & \mu \neq \nu, \end{cases} \tag{4.43}$$

which is equivalent to equation (4.39), and in which $\mu = 1, 2, ..., \mathcal{M}$ refers to a particular moment in the angular expansion of ψ and $(\mathbf{A} \cdot \nabla_{xt}\psi)_\mu$ refers to row μ of the vector arising from the dot product $(\mathbf{A}_{xt} \cdot \nabla_{xt}\psi)$. Row μ of the vectors $||\nabla_{xt}\psi||^2$ and $\nabla_{xt}\psi$ is denoted by $||\nabla_{xt}\psi||^2_\mu$ and $\nabla_{xt}\psi_\mu$ respectively. It is useful to define also the diagonal matrix $\mathbf{B}$ where:

$$\mathbf{B}_{\mu,\nu} = \begin{cases} \frac{(\mathbf{A}\cdot\nabla_{xt}\psi)_\mu}{||\nabla_{xt}\psi||^2_\mu} & \mu = \nu; \\ 0 & \mu \neq \nu, \end{cases} \tag{4.44}$$

i.e. $\mathbf{A}^* = \mathbf{B}(\mathbf{I}\nabla\Psi)$ where $\mathbf{I}\nabla\Psi$ is used to denote an $\mathcal{M} \times \mathcal{M}$ matrix containing row μ of the vector $\nabla\Psi$ at position μ on the main diagonal. The non-linear Petrov-Galerkin approach then uses the above definition of $\mathbf{A}^*$ in the modified form of the differential equation, that is in equation (4.6). Note that the optimal non-linear Petrov-Galerkin method reduces to a standard linear Petrov-Galerkin/SUPG method if one replaces $\mathbf{A}^*$ with $\mathbf{A}$ and $\mathbf{P}^*$ with $\mathbf{P}$ in the modified equation. Modifying the coupled differential equations to obtain a Petrov-Galerkin method, as done for the scalar differential equations in section (4.4) one obtains:

$$(\mathbf{I} - (\nabla_{xt} \cdot \mathbf{A}^*_{xt})^T \mathbf{P}^*_{xt})(\mathbf{A}_{xt} \cdot \nabla_{xt}\Psi + \mathbf{H}\Psi - \mathbf{s}) = \mathbf{0}. \tag{4.45}$$

Equation (4.45) is now tested with a diagonal matrix $\mathbf{N}_{xti}$ which contains the space-time basis function N_{xti} along the main diagonal. One then integrates the equation over a single element V_E and applies integration by parts to the advection terms. This results in:

$$\int_{V_E} dV_{xt} \mathbf{N}_{xti}\mathbf{r} - \oint_{\Gamma_E} d\Gamma_{xt} \mathbf{N}_{xti}(\mathbf{n}_{xt} \cdot \mathbf{A}_{xt})^-(\Psi - \Psi_{bc})$$
$$+ \int_{V_E} dV_{xt}((\nabla_{xt}\mathbf{N}_{xti}) \cdot \mathbf{A}^*_{xt})^T \mathbf{P}^*_{xt}\mathbf{r} + \oint_{\Gamma_E} d\Gamma_{xt} \mathbf{N}_{xti}\mathbf{n}_{xt} \cdot \mathbf{A}^*_{xt}\mathbf{P}^*_{xt}\mathbf{r} = \mathbf{0}, \tag{4.46}$$

with a finite element expansion $\Psi = \sum_{j=1}^{\mathcal{N}} \mathbf{N}_{xtj}\mathbf{\Psi}_j$ (where $\mathbf{\Psi}_j$ is the order $\mathcal{M}$ vector of unknowns at node j) and $\mathbf{r} = \mathbf{A}_{xt} \cdot \nabla_{xt}\Psi + \mathbf{H}\Psi - \mathbf{s}$. Using the eigen-decomposition $\mathbf{n}_{xt} \cdot \mathbf{A}_{xt} = \mathbf{L}_{xt}\mathbf{\Lambda}_{xt}\mathbf{R}_{xt}$ then $(\mathbf{n}_{xt} \cdot \mathbf{A}_{xt})^- = \mathbf{L}_{xt}\mathbf{\Lambda}^-_{xt}\mathbf{R}_{xt}$ with $\mathbf{\Lambda}^-_{xtkk} = min\{0, \mathbf{\Lambda}_{xtkk}\}$. This eigen decomposition enables the boundary condition to be applied to incoming information only. One may apply a zero boundary condition for the residual $\mathbf{r} = \mathbf{0}$, where $\mathbf{0}$ is vector of length $\mathcal{M}$ containing zeroes. This results in:

$$\int_{V_E} dV_{xt}\mathbf{N}_{xti}\mathbf{r} - \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xti}(\mathbf{n}_{xt} \cdot \mathbf{A}_{xt})^-(\Psi - \Psi_{bc}) + \int_{V_E} dV_{xt}((\nabla_{xt}\mathbf{N}_{xti}) \cdot \mathbf{A}^*_{xt})^T\mathbf{P}^*_{xt}\mathbf{r} = \mathbf{0}, \tag{4.47}$$

in which $\mathbf{P}^*_{xt}$ is a function of $\mathbf{A}^*_{xt}$ and the size and shape of the elements, for example:

$$\mathbf{P}^*_{xt} = \frac{1}{4}(|\mathbf{A}^*_{xt} \cdot \nabla_{xt}\mathbf{N}_{xti}|)^{-1}, \tag{4.48}$$

or alternatively, one might use the 2 matrix norm (Euclidean norm or geometric length described by the matrix) and the space-time Jacobian matrix $\mathbf{J}_{xt}$:

$$\mathbf{P}^*_{xt} = \frac{1}{4}(||\mathbf{J}^{-1}_{xt}\mathbf{A}^*_{xt}||_2)^{-1}. \tag{4.49}$$

Since the matrices $\mathbf{A}^*_t$, $\mathbf{A}^*_x$, $\mathbf{A}^*_y$, $\mathbf{A}^*_z$ that go to make up $\mathbf{A}^*_{xt} = (\mathbf{A}^*_t, \mathbf{A}^*_x, \mathbf{A}^*_y, \mathbf{A}^*_z)^T$ are diagonal the matrix $\mathbf{P}^*_{xt}$ is also diagonal. In the traditional Petrov Galerkin method $\mathbf{A}^*_{xt} = \mathbf{A}_{xt}$ in the above and $\mathbf{P}_{xt}$ replaces $\mathbf{P}^*_{xt}$. The finite element space-time Jacobian matrix for 3D time-dependent problems is:

$$\mathbf{J}_{xt} = \begin{pmatrix} \mathbf{I}\frac{\partial t}{\partial t'} & \mathbf{I}\frac{\partial x}{\partial t'} & \mathbf{I}\frac{\partial y}{\partial t'} & \mathbf{I}\frac{\partial z}{\partial t'} \\ \mathbf{I}\frac{\partial t}{\partial x'} & \mathbf{I}\frac{\partial x}{\partial x'} & \mathbf{I}\frac{\partial y}{\partial x'} & \mathbf{I}\frac{\partial z}{\partial x'} \\ \mathbf{I}\frac{\partial t}{\partial y'} & \mathbf{I}\frac{\partial x}{\partial y'} & \mathbf{I}\frac{\partial y}{\partial y'} & \mathbf{I}\frac{\partial z}{\partial y'} \\ \mathbf{I}\frac{\partial t}{\partial z'} & \mathbf{I}\frac{\partial x}{\partial z'} & \mathbf{I}\frac{\partial y}{\partial z'} & \mathbf{I}\frac{\partial z}{\partial z'} \end{pmatrix}, \tag{4.50}$$

where the variables with $\prime$ are the local variables and where $\mathbf{I}$ is the $\mathcal{M} \times \mathcal{M}$ identity matrix in which $\mathcal{M}$ is the number solution variables at each DG node in the mesh. For uniform space-time resolution with a time step size of Δt and an element size

of Δx (in the x-direction), Δy (in the y-direction), Δz (in the z-direction), then:

$$\mathbf{J}_{xt} = \begin{pmatrix} \mathbf{I}\frac{1}{2}\Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\frac{1}{2}\Delta x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}\frac{1}{2}\Delta y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}\frac{1}{2}\Delta z \end{pmatrix}. \tag{4.51}$$

In a similar adjustment to that done for the scalar equation, the value of $\mathbf{P}^*_{xt}$ can be adjusted to ensure that the resulting value of $\mathbf{P}^*_{xt}$ is not so large that there are more particles being transported backwards across an element than forwards, in the resulting discrete system of equations. This is done for the coupled equations by ensuring that:

$$\mathbf{P}^*_{xt} = min\{(\mathbf{H}+\mathbf{E})^{-1}, \frac{1}{4}(||\mathbf{J}^{-1}_{xt}\mathbf{A}^*_{xt}||_2)^{-1}\}, \tag{4.52}$$

in which the diagonal entries of the matrix $\mathbf{H}+\mathbf{E}$ are positive and $\mathbf{E}$ contains small positive numbers to avoid dividing by zero or near zero when one or more of the diagonals of $\mathbf{H}$ is zero or very small e.g. 1×10^{-10}. This applies to voids or very low density materials in the problem.

4.5.1. Diffusion Form

One can alternatively use a stabilisation in the coupled equations that is of diffusion form, rather than the Petrov-Galerkin form. As discussed for the scalar equations, this might offer superior convergence rates of the non-linear term in problems where a residual based non-linear term is too sensitive to oscillations in the solution. A diffusion form might damp this behaviour and converge rapidly. Writing the coupled non-linear equations with this form of stabilisation, one has:

$$\int_{V_E} dV_{xt}\mathbf{N}_{xti}\mathbf{r} - \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xti}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}(\Psi-\Psi_{bc}) + \int_{V_E} dV_{xt}(\nabla_{xt}\mathbf{N}_{xti})^T\mathbf{K}\nabla_{xt}\Psi = \mathbf{0}. \tag{4.53}$$

Equation (4.53) may alternatively be written in a form where one applies integration by parts to the transport terms once:

$$\int_{V_E} dV_{xt}\mathbf{N}_{xti}(\mathbf{H}\Psi - \mathbf{s}) - \int_{V_E} dV_{xt}(\nabla_{xt}\cdot(\mathbf{N}_{xti}\mathbf{A}_{xt}))\Psi + \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xti}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}\Psi_{bc} + \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xti}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{+}\Psi + \int_{V_E} dV_{xt}(\nabla_{xt}\mathbf{N}_{xti})^{T}\mathbf{K}\nabla_{xt}\Psi = \mathbf{0}, \tag{4.54}$$

in which the $\mathcal{M}\times\mathcal{M}$ diagonal matrix containing the diffusion coefficients is:

$$\mathbf{K} = \mathbf{V}(\mathbf{A}_{xt}\cdot\nabla_{xt}\Psi)\mathbf{P}^{*}_{xt}\mathbf{V}(||\nabla_{xt}\Psi||_2^2)^{-1}\mathbf{V}(\mathbf{r}). \tag{4.55}$$

The resulting diagonal matrix $\mathbf{K}$ can be modified to ensure non-negative diffusion by setting any of its negative entries to zero or taking their absolute values. Alternatively one can work with the residual only, by replacing $\mathbf{A}_{xt}\cdot\nabla_{xt}\Psi$ with the residual $\mathbf{r}$, which results in:

$$\mathbf{K} = \mathbf{V}(\mathbf{r})^{T}\mathbf{P}^{*}_{xt}\mathbf{V}(||\nabla_{xt}\Psi||_2^2)^{-1}\mathbf{V}(\mathbf{r}). \tag{4.56}$$

This is always positive because $\mathbf{P}^{*}_{xt}$ is positive semi-definite (as well as diagonal) and in which $\mathbf{V}(\mathbf{r})$ is the diagonal matrix containing the residual of the governing equations on its diagonal. Equation (4.56) for the diffusivity can be derived by re-defining $\mathbf{A}^{*}_{xt}$ in equation (4.39) to:

$$\mathbf{A}^{*}_{xt} = \mathbf{V}(\mathbf{r})\mathbf{V}(||\nabla_{xt}\Psi||_2^2)^{-1}\nabla_{xt}\Psi. \tag{4.57}$$

4.5.2. Time-Dependence

As done for the scalar equations, one can assume the variation in time is discretised using the two level θ-method:

$$\mathbf{r} = \mathbf{A}_t\frac{\Psi^{n+1}-\Psi^{n}}{\Delta t} + \mathbf{A}\cdot\nabla\Psi^{n+\theta} + \mathbf{H}\Psi^{n+\theta} - \mathbf{s}^{n+\theta}, \tag{4.58}$$

with $\mathbf{A} = (\mathbf{A}_x \quad \mathbf{A}_y \quad \mathbf{A}_z)^T$ and $\Psi^{n+\theta} = \Theta\Psi^{n+1} + (\mathbf{I} - \Theta)\Psi^n$ in which Θ is a diagonal matrix containing the time stepping parameters and also defining:

$$\nabla_{xt}\Psi = \left(\frac{\Psi^{n+1} - \Psi^n}{\Delta t}, \ \left(\nabla \Psi^{n+\theta} \right)^T \right)^T . \tag{4.59}$$

Using this definition (equation (4.59)) enables the application of the mechanics of space-time discretisation, developed here, to be applied to the coupled equations. For example:

$$\mathbf{A}^*_{xt} = (\mathbf{A}^{*T}_t, \ \mathbf{A}^{*T})^T = \mathbf{V}(\mathbf{A}_{xt} \cdot \nabla_{xt}\Psi)\mathbf{V}(||\nabla_{xt}\Psi||^2_2)^{-1}\nabla_{xt}\Psi \tag{4.60}$$

and

$$\mathbf{P}^*_{xt} = min\{(\mathbf{H} + \mathbf{E})^{-1}, \frac{1}{4}(||\mathbf{J}^{-1}\mathbf{A}^*||_2)^{-1}\}. \tag{4.61}$$

By applying diffusion only in Cartesian space the stabilised discrete equations in diffusion form can be written:

$$\int_{V_E} dV \mathbf{N}_i \mathbf{r} - \oint_{\Gamma_E} d\Gamma \mathbf{N}_i (\mathbf{n} \cdot \mathbf{A})^- (\Psi^{n+\theta} - \Psi^{n+\theta}_{bc}) + \int_{V_E} dV (\nabla \mathbf{N}_i)^T \mathbf{K} \nabla \Psi^{n+1} = \mathbf{0}, \tag{4.62}$$

or in a form where we apply integration by parts once to the transport terms:

$$\int_{V_E} dV \mathbf{N}_i (\mathbf{A}_t \frac{\Psi^{n+1} - \Psi^n}{\Delta t} + \mathbf{H}\Psi^{n+\theta} - \mathbf{s}^{n+\theta}) - \int_{V_E} dV \nabla \cdot (\mathbf{N}_i \mathbf{A}) \Psi^{n+\theta} + \oint_{\Gamma_E} d\Gamma \mathbf{N}_i (\mathbf{n} \cdot \mathbf{A})^- \Psi^{n+\theta}_{bc} + \oint_{\Gamma_E} d\Gamma \mathbf{N}_i (\mathbf{n} \cdot \mathbf{A})^+ \Psi^{n+\theta} + \int_{V_E} dV (\nabla \mathbf{N}_i)^T \mathbf{K} \nabla \Psi^{n+1} = \mathbf{0}. \tag{4.63}$$

4.5.3. Optimal Petrov-Galerkin Methods

In order to apply the optimal Petrov-Galerkin method it is necessary to diagonalise the $\mathbf{H}'$ in the governing equation defined by its residual:

$$\mathbf{r}' = \mathbf{A}'_{xt} \cdot \nabla_{xt}\Psi' + \mathbf{H}'\Psi' - \mathbf{s}', \tag{4.64}$$

using the eigenvalue decomposition:

$$\mathbf{H}' = \mathbf{R}_H^T \mathbf{\Lambda}_H \mathbf{R}_H, \tag{4.65}$$

in which $\mathbf{\Lambda}_H = \mathbf{H}$ is a diagonal matrix and $\mathbf{R}_H^T \mathbf{R}_H = \mathbf{I}$. Defining $\mathbf{A}_{xt} = \mathbf{R}_H^{-T} \mathbf{A}'_{xt} \mathbf{R}_H^{-1}$ then multiplying equation (4.64) by $\mathbf{R}_H^{-T}$ using $\Psi = \mathbf{R}_H \Psi'$ and assuming the material properties are uniform the following matrix equation results:

$$\mathbf{A}_{xt} \cdot \nabla_{xt} \Psi + \mathbf{H} \Psi = \mathbf{s}. \tag{4.66}$$

One will note that equation (4.66) is simply equal to $\mathbf{R}_H^T r'$ in which r' is the residual of the original equation. Thus it is a very straightforward reverse mapping operation to return the equation to the original space. The non-dimensional absorption coefficient is contained in the diagonal matrix:

$$\hat{\mathbf{H}} = \left(||\mathbf{J}_{xt}^{-1} \mathbf{A}_{xt}^*||_2 \right)^{-1} \mathbf{H}, \tag{4.67}$$

and the stabilisation matrix that appears in the non-linear term defined to be:

$$\mathbf{P}^* = \underline{\underline{\alpha}}(\hat{\mathbf{H}}) \left(||\mathbf{J}_{xt}^{-1} \mathbf{A}_{xt}^*||_2 \right)^{-1}, \tag{4.68}$$

in which the diagonal matrix $\underline{\underline{\alpha}}(\hat{\mathbf{H}})$ is such that $\underline{\underline{\alpha}}(\hat{\mathbf{H}})_{\mu\mu} = \alpha(\hat{\sigma}_\mu)$ with $\hat{\sigma}_\mu = \hat{\mathbf{H}}_{\mu\mu}$. If one now maps the equations back to the original system (equation (4.64)) and then inserts them into the discretisation, as explained in the previous sections, the result is:

$$\begin{aligned} \int_{V_E} dV_{xt} \mathbf{N}_{xti} \mathbf{r}' - \oint_{\Gamma_E} d\Gamma_{xt} \mathbf{N}_{xti} (\mathbf{n}_{xt} \cdot \mathbf{A}'_{xt})^- (\Psi' - \Psi'_{bc}) \\ + \int_{V_E} dV_{xt} (\nabla_{xt} \mathbf{N}_{xti})^T \mathbf{R}_H \mathbf{K} \mathbf{R}_H^T \nabla_{xt} \Psi' = \mathbf{0}. \end{aligned} \tag{4.69}$$

The viscosity now becomes a non-diagonal $\mathcal{M} \times \mathcal{M}$ matrix: $\mathbf{R}_H \mathbf{K} \mathbf{R}_H^T$.

4.6. Positive Diffusion

The optimal non-linear PG method was found to work extremely well in its most basic form for the S_N and P_N equations. Symmetry issues were realised, however, when an attempt was made to apply the scheme to LW_N wavelet approximations.

The origin of this asymmetry is illustrated when one writes the optimal non-linear PG equations in diffusion form. Considering the equations in diffusion form also offers a simple remedy for the asymmetry. Implementing the method in diffusion form therefore allows it to be applied to any angular discretisation. Non-linearity is introduced via the gradient-informed modification to the advection velocity. Integrating the non-linear term by parts yields:

$$\int_{V_E} dV \nabla \mathbf{N}_i \cdot \mathbf{A}^* \mathbf{P}^* \mathscr{R} = \int_{V_E} dV \nabla \mathbf{N}_i \cdot \mathbf{G} \left(\mathbf{I} \nabla \Psi \right), \tag{4.70}$$

where the notation $(\mathbf{I}\nabla\Psi)$ is used to describe an $\mathcal{M} \times \mathcal{M}$ diagonal matrix with $\nabla\Psi_m = (\frac{\partial \Psi_m}{\partial x}, \frac{\partial \Psi_m}{\partial y}, \frac{\partial \Psi_m}{\partial z}, \frac{\partial \Psi_m}{\partial t})^T$ at position m on the main diagonal. $\mathbf{G}$ is a diagonal matrix defined to be:

$$\mathbf{G} = \mathbf{B}\mathbf{P}^* \mathscr{R}, \tag{4.71}$$

where $\mathbf{B}$ is defined as in equation (4.44). One may write:

$$\mathbf{G} = \mathbf{D}\mathbf{P}^*, \tag{4.72}$$

in which matrix $\mathbf{D}$ is defined to be:

$$\mathbf{D} = \mathbf{B}(\mathbf{I}\mathscr{R}), \tag{4.73}$$

where $\mathbf{I}\mathscr{R}$ has been used to denote a diagonal matrix with row μ of vector $\mathscr{R}$ at position μ on the main diagonal. Thus $\mathbf{D}$ will be a diagonal matrix. For the discrete ordinates and spherical harmonics approximations, $\mathbf{H}$ is diagonal. This results in $\mathbf{P}^*$ being a diagonal matrix that is positive semi-definite. Hence matrix $\mathbf{G} = \mathbf{D}\mathbf{P}^*$ will be symmetric. However, for wavelet approximations $\mathbf{H}$ is not diagonal. This results in $\mathbf{P}^*$ being a full matrix and hence the product $\mathbf{D}\mathbf{P}^*$ will be asymmetric. One can make $\mathbf{G}$ symmetric for the wavelet method by using the alternative definition:

$$\mathbf{G} = \mathbf{D}^{\frac{1}{2}} \mathbf{P}^* \mathbf{D}^{\frac{1}{2}}, \tag{4.74}$$

in place of $\mathbf{G} = \mathbf{D}\mathbf{P}^*$. Equation (4.74) yields a symmetric matrix, despite $\mathbf{P}^*$ being non-diagonal. However, this requires positivity in $\mathbf{D}$ which can be guaranteed

by using:

$$\mathbf{A}^* = \frac{-sign(\mathscr{R})|\mathbf{A}\cdot\nabla\Psi|\nabla\Psi}{||\nabla\Psi||^2}, \tag{4.75}$$

in place of equation (4.39).

4.7. Void Treatment

In a material, the optimal non-linear PG approach discussed relies on a Cholesky factorisation of the scattering removal operator to reliably form a diagonal system of equations in Riemann space. That is to use $\mathbf{H}_s = \mathbf{G}^T\mathbf{G}$ in which $\mathbf{H}_s = \mathbf{L}_s\mathbf{H}\mathbf{R}_s$ is the mapping of the scattering-removal matrix operator into the Riemann space of $\mathbf{A}_s$, and $\mathbf{G}^T$ and $\mathbf{G}$ are the Cholesky factors of $\mathbf{H}_s$. These Cholesky factors are defined only if $\mathbf{H}_s$ has a positive non-zero eigenstructure. The strength of this approach is that it allows the optimal scheme to be applied within any angular discretisation framework, as a diagonal system can easily be formed. In a void, the problem reduces to pure advection comprising only the streaming term. The $\mathbf{H}$ term is absent from the equation and so the $\mathbf{H}_s$ Riemann term is not present in the one-dimensional diagonal equation. Consequently, the system along the streamline in a void is simply:

$$\mathbf{\Lambda}_s\frac{\partial\Psi_s}{\partial s} + \alpha h\mathbf{\Lambda}_s\frac{\partial}{\partial s}(\mathbf{\Lambda}_s\frac{\partial\Psi_s}{\partial s}) = \mathbf{0}, \tag{4.76}$$

where h is the length scale, typically $h = \frac{\Delta x}{4}$ or $h = \frac{\Delta x}{2}$ and Δx is the width of the element. α is the optimal coefficient for the artificial dissipation in a void. By examining the DG stencil of equation (4.76) using this choice of length scale, one finds that $\alpha = \mathbf{I}$ matches the analytical solution at the outlet node of the element. Selecting this choice of α in elements that contain a void, while using the previous definition elsewhere, one has an optimal non-linear PG scheme that can stabilise transport solutions across all regimes.

4.8. Length Scale

Conventional Petrov-Galerkin formulations use $\frac{\Delta x}{2}$ as the characteristic length scale, in which Δx is the width of the finite element in space. This correctly centres the equation residual at the centre of mass (COM) of the basis function, for

continuous finite element representations. In the present work, where discontinuous finite elements are used to formulate the space-time discretisation, the COM of the basis function is centred a distance $\frac{\Delta x}{4}$ from the upwind boundary of the element. It is recommended therefore, that $h = \frac{\Delta x}{4}$ be used with the optimal non-linear PG method instead of $h = \frac{\Delta x}{2}$. This translates the equation residual along the gradient of the basis function adding additional upwinding to the optimal non-linear PG finite element stencil. This correctly positions the residual calculation at the COM of the optimal non-linear PG discretisation. This is important for ensuring the method uses the correct amount of upwinding. The finite element stencil for the standard DG method without using any added dissipation is shown to be:

$$\begin{bmatrix} -1 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix}, \tag{4.77}$$

while the optimal non-linear stencil, using $h = \frac{\Delta x}{4}$ as the length scale, is found to be:

$$\begin{bmatrix} -1 & \frac{3}{4} & \frac{1}{4} & 0 & 0 \\ 0 & -\frac{3}{4} & \frac{3}{4} & 0 & 0 \end{bmatrix}. \tag{4.78}$$

Examining these two stencils, one can see that using $h = \frac{\Delta x}{4}$ for the length scale in the optimal non-linear PG discretisation amounts to increasing the upwinding by 50%, with respect to standard DG. Interestingly, the downwind balance remains unaltered. Note this is akin to shifting the residual calculation upwind along the gradient by using $(x_0 - \frac{\Delta x}{4})$ in the Taylor expansion of $\mathscr{R}$, about the origin x_0 of the discontinuous finite element stencil. Note that the stencil for standard PG methods using a length scale of $\frac{\Delta x}{2}$ is:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{bmatrix}. \tag{4.79}$$

4.9. Non-Linear Petrov-Galerkin Sub-Grid-Scale Methods

Sub-Grid-Scale (SGS) methods decompose the solution into an continuous part Ψ^{cty} and a SGS internal to each element part Ψ^{SGS} with basis functions $\mathbf{N}_{xt_i}^{cty}$ and $\mathbf{N}_{xt_i}^{SGS}$ respectively and thus:

$$\Psi = \sum_{j=1}^{\mathcal{N}^{cty}} \mathbf{N}_{xt_j}^{cty} \Psi_j^{cty} + \sum_{j=1}^{\mathcal{N}^{SGS}} \mathbf{N}_{xt_j}^{SGS} \Psi_j^{SGS}. \tag{4.80}$$

The basis functions $\mathbf{N}_{xt_j}^{SGS}$ are typically discontinuous between elements and so SGS representations have much in common with DG methods in fact they can be shown to be equivalent in 1D. However, one of the advantages of SGS methods is that the inner element solutions can be eliminated from the system of equations with static condensation which reduces the number of unknowns to solve for greatly - reducing them down to continuous node unknowns. The element boundary conditions on the SGS solution Ψ^{SGS} is zero for incoming element information and for incoming and outgoing information on the boundary of the solution domain. The boundary conditions on the domain were chosen so as to form a non-singular system of equations even though the basis functions do not necessarily form an independent set of functions. To form the discretised equations the Petrov-Galerkin differential equation:

$$(\mathbf{I} - (\nabla_{xt} \cdot \mathbf{A}_{xt}^*)^T \mathbf{P}_{xt}^*)\mathbf{r} = \mathbf{0}, \tag{4.81}$$

is tested with both sets of basis functions:

$$\begin{aligned} \int_V dV_{xt} \mathbf{N}_{xt_i}^{cty} \mathbf{r} - \oint_\Gamma d\Gamma_{xt} \mathbf{N}_{xt_i}^{cty} (\mathbf{n}_{xt} \cdot \mathbf{A}_{xt})^- (\Psi^{cty} - \Psi_{bc}) \\ + \int_V dV_{xt} ((\nabla_{xt} \mathbf{N}_{xt_i}^{cty}) \cdot \mathbf{A}_{xt}^*)^T \gamma \mathbf{P}_{xt}^* \mathbf{r} \\ - \oint_\Gamma d\Gamma_{xt} \mathbf{N}_{xt_i}^{cty} \mathbf{n}_{xt} \cdot \mathbf{A}_{xt}^* \gamma \mathbf{P}_{xt}^* \mathbf{r} = \mathbf{0}, \end{aligned} \tag{4.82}$$

and :

$$\int_{V_E} dV_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{r} - \oint_{\Gamma} d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}(\Psi^{cty}-\Psi_{bc})$$
$$+\oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}\Psi^{SGS} + \oint_{\Gamma} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{+}\Psi^{SGS}$$
$$+\int_{V_E} dV_{xt}((\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})\cdot\mathbf{A}_{xt}^{*})^{T}\mathbf{P}_{xt}^{*}\mathbf{r}$$
$$-\oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{n}_{xt}\cdot\mathbf{A}_{xt}^{*}\mathbf{P}_{xt}^{*}\mathbf{r}=\mathbf{0}. \quad (4.83)$$

Applying a zero residual $\mathbf{r}=0$ on the boundaries of the elements results in:

$$\int_{V} dV_{xt}\mathbf{N}_{xt_i}^{cty}\mathbf{r} - \int_{\Gamma} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}(\Psi^{cty}-\Psi_{bc})$$
$$+\int_{V} dV_{xt}((\nabla_{xt}\mathbf{N}_{xt_i}^{cty})\cdot\mathbf{A}_{xt}^{*})^{T}\gamma\mathbf{P}_{xt}^{*}\mathbf{r}=\mathbf{0}, \quad (4.84)$$

and :

$$\int_{V_E} dV_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{r} - \oint_{\Gamma} d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}(\Psi^{cty}-\Psi_{bc})$$
$$+\oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{-}\Psi^{SGS} + \oint_{\Gamma} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^{+}\Psi^{SGS}$$
$$+\int_{V_E} dV_{xt}((\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})\cdot\mathbf{A}_{xt}^{*})^{T}\mathbf{P}_{xt}^{*}\mathbf{r}=\mathbf{0}, \quad (4.85)$$

in which $\gamma = 1$ applies Petrov-Galerkin weighting to both the continuous test functions $\mathbf{N}_{xt_i}^{cty}$ and the SGS test functions $\mathbf{N}_{xt_i}^{SGS}$. To apply Petrov-Galerkin weighting only to the SGS test functions or inner element solution then $\gamma = 0$.

The corresponding diffusion form of the discretised equations is:

$$\int_V dV_{xt}\mathbf{N}_{xt_i}^{cty}\mathbf{r} - \oint_\Gamma d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^-(\Psi^{cty}-\Psi_{bc}) + \int_V dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{cty})^T\mathbf{K}\nabla_{xt}\Psi^{cty} + \int_V dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{cty})^T\mathbf{K}\nabla_{xt}\Psi^{SGS} - \oint_{\Gamma_E} d\Gamma_{xt}(\mathbf{n}_{xt}\cdot\mathbf{K}\nabla_{xt}\mathbf{N}_{xt_i}^{cty})\Psi^{SGS} = \mathbf{0}, \quad (4.86)$$

and:

$$\int_{V_E} dV_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{r} - \oint_\Gamma d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^-(\Psi^{cty}-\Psi_{bc}) + \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^-\Psi^{SGS} + \oint_\Gamma d\Gamma_{xt}\mathbf{N}_{xt_i}^{cty}(\mathbf{n}_{xt}\cdot\mathbf{A}_{xt})^+\Psi^{SGS} + \int_V dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})^T\mathbf{K}\nabla_{xt}\Psi^{cty} - \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{n}_{xt}\cdot\mathbf{K}\nabla_{xt}\Psi^{cty} + \int_V dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})^T\mathbf{K}\nabla_{xt}\Psi^{SGS} - \oint_{\Gamma_E} d\Gamma_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{n}_{xt}\cdot\mathbf{K}\nabla_{xt}\Psi^{SGS} - \oint_{\Gamma_E} d\Gamma_{xt}(\mathbf{n}_{xt}\cdot\mathbf{K}\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})\Psi^{SGS} = \mathbf{0}. \quad (4.87)$$

The last three terms in equation (4.87) (the discretisation of SGS diffusion) can also be represented by:

$$-\int_V dV_{xt}\mathbf{N}_{xt_i}^{SGS}\nabla_{xt}(\mathbf{K}^{\frac{1}{2}}\mathbf{q}^{SGS}), \quad (4.88)$$

with $\mathbf{q}^{SGS}$ determined from:

$$\int_V dV_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{q}^{SGS} = -\int_V dV_{xt}(\nabla_{xt}(\mathbf{K}^{\frac{1}{2}}\mathbf{N}_{xt_i}^{SGS}))\Psi^{SGS}. \quad (4.89)$$

The last terms in equations (4.86) and (4.87) have been introduced to help enforce

$\Psi^{SGS} = \mathbf{0}$, on the element boundaries Γ_E, through:

$$\int_V dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{cty})^T\mathbf{K}\nabla_{xt}\Psi^{SGS}$$
$$= \int_{V_E} dV_{xt}(\nabla_{xt}\mathbf{N}_{xt_i}^{cty})^T\mathbf{K}\nabla_{xt}\Psi^{SGS}$$
$$- \int_{\Gamma_E} d\Gamma_{xt}(\mathbf{n}_{xt}\cdot\mathbf{K}\nabla_{xt}\mathbf{N}_{xt_i}^{cty})(\Psi^{SGS} - \Psi_{bc}^{SGS}), \tag{4.90}$$

and with $\Psi_{bc}^{SGS} = \mathbf{0}$. The surface integral around each element in equation (4.89) has been set to zero to enforce $\Psi^{SGS} = \mathbf{0}$ on the element boundaries Γ_E. One can ignore all surface integrals in the above which involve the diffusion coefficient $\mathbf{K}$ as this effectively enforces $\mathbf{r} = \mathbf{0}$ on the element boundaries Γ_E. If the residual $\mathbf{r}$ used to calculate the diffusion coefficient $\mathbf{K}$ contains second order or higher derivatives then it is suggested that they may be evaluated using the FEM basis functions so that for linear elements and second order derivatives there is zero contribution from them in this residual. Since the matrices local to each element may be relatively small one can manipulate their eigen structure in order to construct stabilisation methods. For example, if one were to stabilise within an element ($\gamma = 0$), the following can be defined:

$$\mathbf{B}^{SGS}{}_{ij} = \int_{V_E} dV\mathbf{N}_{xt_i}^{SGS}\mathbf{A}_{xt}^{*}\cdot\nabla\mathbf{N}_{xt_j}^{SGS}. \tag{4.91}$$

and the last term of equation (4.85), that is $\int_{V_E} dV_{xt}((\nabla_{xt}\mathbf{N}_{xt_i}^{SGS})\cdot\mathbf{A}_{xt}^{*})^T\mathbf{P}_{xt}^{*}\mathbf{r}$, can be replaced by:

$$(\mathbf{B}^{SGS^T}\mathbf{O}^{SGS^{-1}}\mathbf{l}^{SGS})_i, \tag{4.92}$$

with:

$$\mathbf{l}_i^{SGS} = \int_{V_E} dV_{xt}\mathbf{N}_{xt_i}^{SGS}\mathbf{r}, \tag{4.93}$$

$$\mathbf{O}^{SGS^{-1}} = |\mathbf{B}^{SGS}|^{-1} = \mathbf{R}_{B^{SGS}}|\mathbf{\Lambda}_{B^{SGS}}|^{-1}\mathbf{L}_{B^{SGS}}, \tag{4.94}$$

$\mathbf{R}_{B^{SGS}}, \mathbf{L}_{B^{SGS}}$ are matrices of right and left eigen-vectors of $\mathbf{B}^{SGS}$ and $\mathbf{\Lambda}_{B^{SGS}}$ is the matrix of eigen-values of $\mathbf{B}^{SGS}$. A possible alternative to $\mathbf{O}^{SGS^{-1}}$ is defined

by equation (4.94) which would ensure positive diffusion is:

$$\mathbf{O}^{SGS^{-1}} = (\mathbf{B}^{SGS^T}\mathbf{B}^{SGS})^{-\frac{1}{2}}. \tag{4.95}$$

4.10. Numerical Examples

In this section results are presented for the non-linear optimal solver algorithm using different angular schemes. These include the spherical harmonics method and the discrete ordinates method. Linear optimal methods have already been shown to work extremely well for steady-state problems. It is found, however, they can fail when applied to transient problems. An example of such failure is shown in the present section. Introducing non-linearity removes this defect, which is demonstrated.

4.10.1. Problem 1: Time-Dependent Absorber

Figure (4.2) illustrates in the solution domain of problem 1, the time-dependent absorber. The materials are shown in table (4.1). A cross-section of 7cm^{-1} is used throughout the 12cm x 12cm spatial part of the domain. An imposed radiation source is located in the range 5cm $\leq$ x $\leq$ 8cm, 5 cm $\leq$ y $\leq$ 8 cm. The cross-section inside the source is also 7cm^{-1}. The problem runs for 12s for which 12 time-steps are used, thus a 1728 element mesh is required for this configuration. The cross-section is significant, and combined with the relatively low spatial and temporal resolution makes this a very demanding problem for the standard DG method. The original optimal method, shown to work well for steady-state solutions, is shown to fail in figure (4.3). This plot comprises line-outs of the scalar flux solution to problem 1. These line-outs are through the centre of the mesh, parallel to the x,y and temporal axes. The DG solution shows severe oscillation due to the inadequate spatial and temporal resolution. Note that the Z direction represents the time direction in this graph. The linear optimal method degrades the solution to this problem considerably, introducing large oscillations in the time direction that are not seen on the standard DG solution. Introducing non-linearity totally removes these oscillations, and the resulting scheme stabilises the standard DG result very effectively. This is demonstrated in figure (4.4) for (a,b) a P_3 approximation and (c,d) an S_4 approximation. The non-linear PG scheme performs consistently well for both angular approximations, fully stabilising the FE solution. The exact solu-

tion is plotted for benchmarking. This shows that the non-linear method improves accuracy as well as robustness.

4.10.2. Problem 2: Void

The non-linear PG method offers an effective remedy for oscillation in voids, where other schemes may perform poorly. To demonstrate the void treatment using the non-linear PG scheme, the domain illustrated in figure (4.5) was used. This comprised a 12 cm $\times$ 12 cm region driven along the left and lower boundaries with a flux of 1.0 $\mathrm{ncm}^{-2}\mathrm{s}^{-1}$, using pure advection to transport the ray at $45\mathrm{deg}$ across the mesh. Scalar flux line-outs across the centre (Line-Out A) and along the top edge (Line-Out B) of the domain are presented for each method on coarse and fine spatial grids in figure (4.6). For these calculations, an S_2 angular approximation was used, producing a well defined ray. The sharp front of the ray is clearly visible on both grids for both methods, and places strain on the spatial approximation causing additional oscillation. These oscillations are smoothed extremely effectively by the non-linear PG method using just 20 non-linear matrix iterations, along both line-outs. Note that even on a fine mesh, the DG result exhibits significant oscillation. The non-linear PG method, however, is free from this oscillation that arises from inadequacies of the DG spatial differencing scheme.

4.10.3. Problem 3: Heavy Absorber Problem

The third test case is illustrated in figure (4.7). This problem cannot be stabilised by conventional linear methods, since the oscillations occur in the direction of the solution gradient. The problem comprises a source region in the range 1cm $\leq$ x $\leq$ 2cm, 1 cm $\leq$ y $\leq$ 2 cm. This is located inside a heavy absorber of range 0cm $\leq$ x $\leq$ 2cm, 1 cm $\leq$ y $\leq$ 2 cm. The materials used are shown in table (4.2). Results for the DG and non-linear PG schemes are illustrated in figure (4.8). Four line-outs are shown. Line-out A in figure (4.8a) is through the centre of the domain in the x direction, with line-out B in figure (4.8b) showing details at the corner of the peak. Line-out C in figure (4.8c) is through the centre of the domain in the y direction, with details at the corner of the peak shown in figure (4.8d) for this direction. The DG scheme produces oscillation all the way round the edge of the source peak. These are clearly visible in the line-outs. The non-linear PG removes these oscillations from the transport solution very effectively in this problem. This is because the dissipation is applied in the gradient direction.

4.10.4. Problem 4: Pure Advection Problem

This problem tests the non-linear Petrov-Galerkin method in an advection process. This uses the method described in section (4.4) to solve $\frac{\partial \psi}{\partial t} + \frac{\partial \psi}{\partial x} = 0$ for0.3 advection time units. This is done using a small time step size of $\Delta t = 0.001$. The results are plotted in figure (4.10).

4.10.5. Convergence

An attempt was made to improve the convergence rate of the non-linear PG iteration. This is achieved by slowing down the rate of convergence by combining the solution Ψ_k from the current iteration k with the old solution Ψ_{k-1} from the previous iteration k-1. Each of these two solutions is weighted with parameter $0.0 \leq \omega \leq 1.0$. The matrix solve is then advanced using the variable $\Psi_{k+1} = \omega\Psi_k + (1-\omega)\Psi_{k-1}$. Figure (4.9) shows the effect that different relaxation parameters ω have on the non-linear PG matrix solve, where the 2-norm is plotted against iteration number. Convergence is approached more smoothly with a greater amount of relax, although many more iterations result. Reducing the amount of relax heavily reduces the required number of iterations to achieve a given error however the approach is less smooth. For practical simulations, $\omega = 0.5$ is recommended for the class of problems in this report. It is sometimes beneficial to use an adaptive relax. This is implemented by reducing the value of ω in response to a change in the behaviour of the solution during the iteration process. In the current work, the value of ω is reduced by a factor 2 each time the behaviour of the 2-norm of the solution changes between non-linear matrix iterations. Another alternative is to optimise the relaxation parameter, by solving $\Psi_{k+1} = \omega\Psi_k + (1-\omega)\Psi_{k-1}$ for the value of ω that minimises the 2-norm of the system. Such a topic is left for the subject of future work.

4.11. Conclusions

A new optimal non-linear discontinuous Petrov-Galerkin method is presented that is extremely effective for removing oscillations in the solution to the Boltzmann Transport Equation in transient and steady-state radiation transport applications. The method applies artificial dissipation internal to an element via a novel finite element Riemann approach, in the direction of the solution gradient. Conventional linear stabilisation methods, such as SUPG, apply dissipation in the direction of

the streamline rather than the direction of the solution gradient. Such methods use the same length scale throughout the solution domain, and are shown not to remove oscillation from the solution gradient direction. There are a class of problems that exhibit additional oscillation in the gradient direction that therefore cannot be stabilised by conventional streamline upwind methods. By using a gradient-informed scaling of the advection velocity, the new method is able to smooth this type of oscillation extremely effectively in a variety of radiation transport problems. Results have been presented for a set of demanding steady-state and time-dependent test problems. Methods for improving the convergence rate of the non-linear iteration used in the optimal non-linear PG method have been presented. This includes an adaptive relaxation method that dramatically reduces the number of iterations needed to achieve highly converged solutions to the problem.

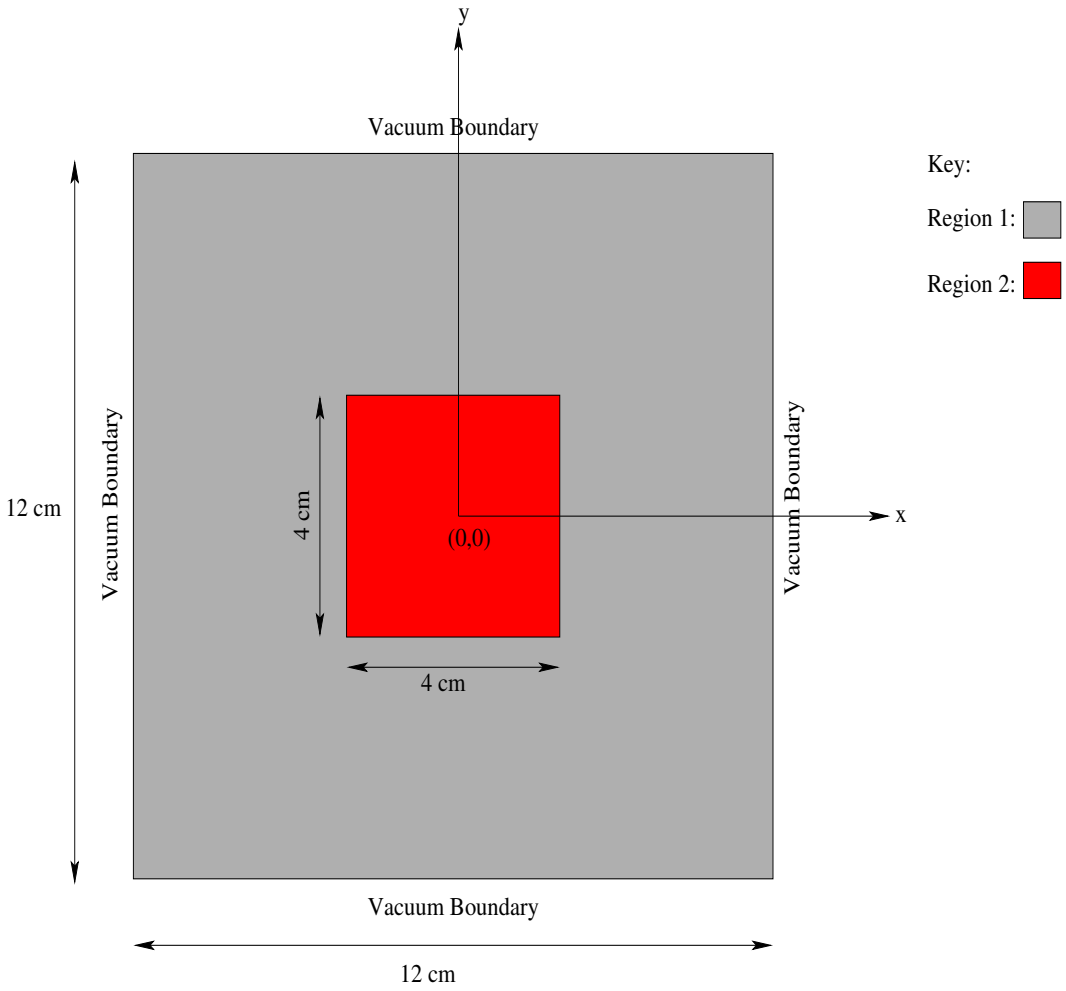
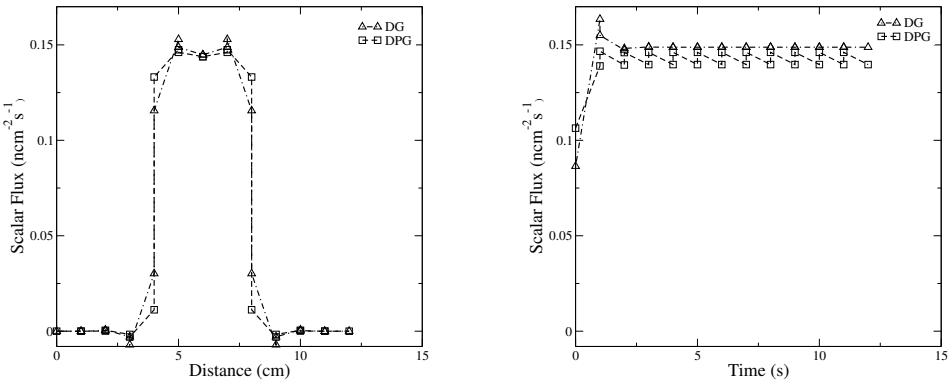


Figure 4.2.: The time-dependent absorber problem.

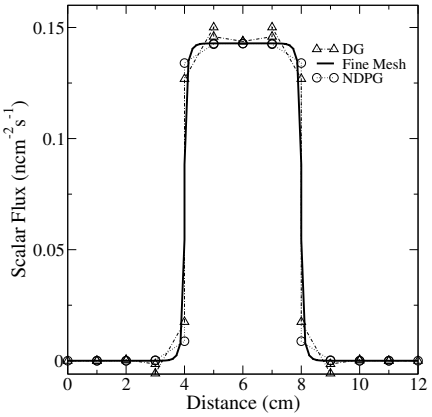
Material Region	Description	Source (n s^{-1})	σ_t (cm^{-1})	σ_s (cm^{-1})
1	Source	1.0	7.0	0.0
2	Absorber	0.0	7.0	0.0

Table 4.1.: Definition of materials in the time-dependent absorber problem.

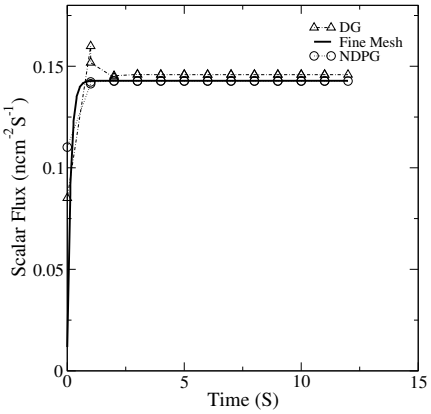


(a) Line-Out A (b) Line-Out B

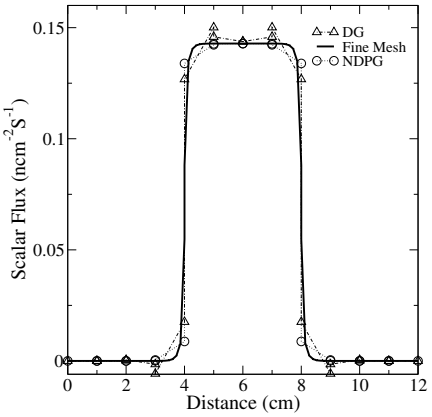
Figure 4.3.: DG and optimal non-linear Scalar flux solutions to the time-dependent absorber problem (a) along the spatial directions and (b) through time.



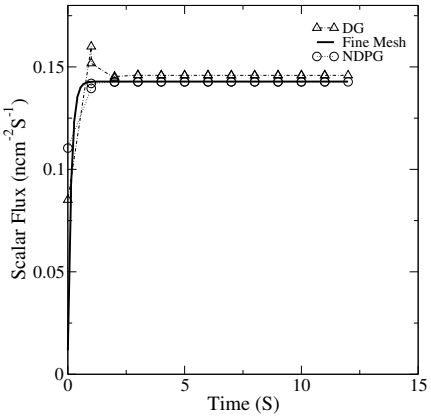
(a) P_3 Line-Out A



(b) P_3 Line-Out B



(c) S_4 Line-Out A



(d) S_4 Line-Out B

Figure 4.4.: The time-dependent absorber problem optimal non-linear and DG solutions.

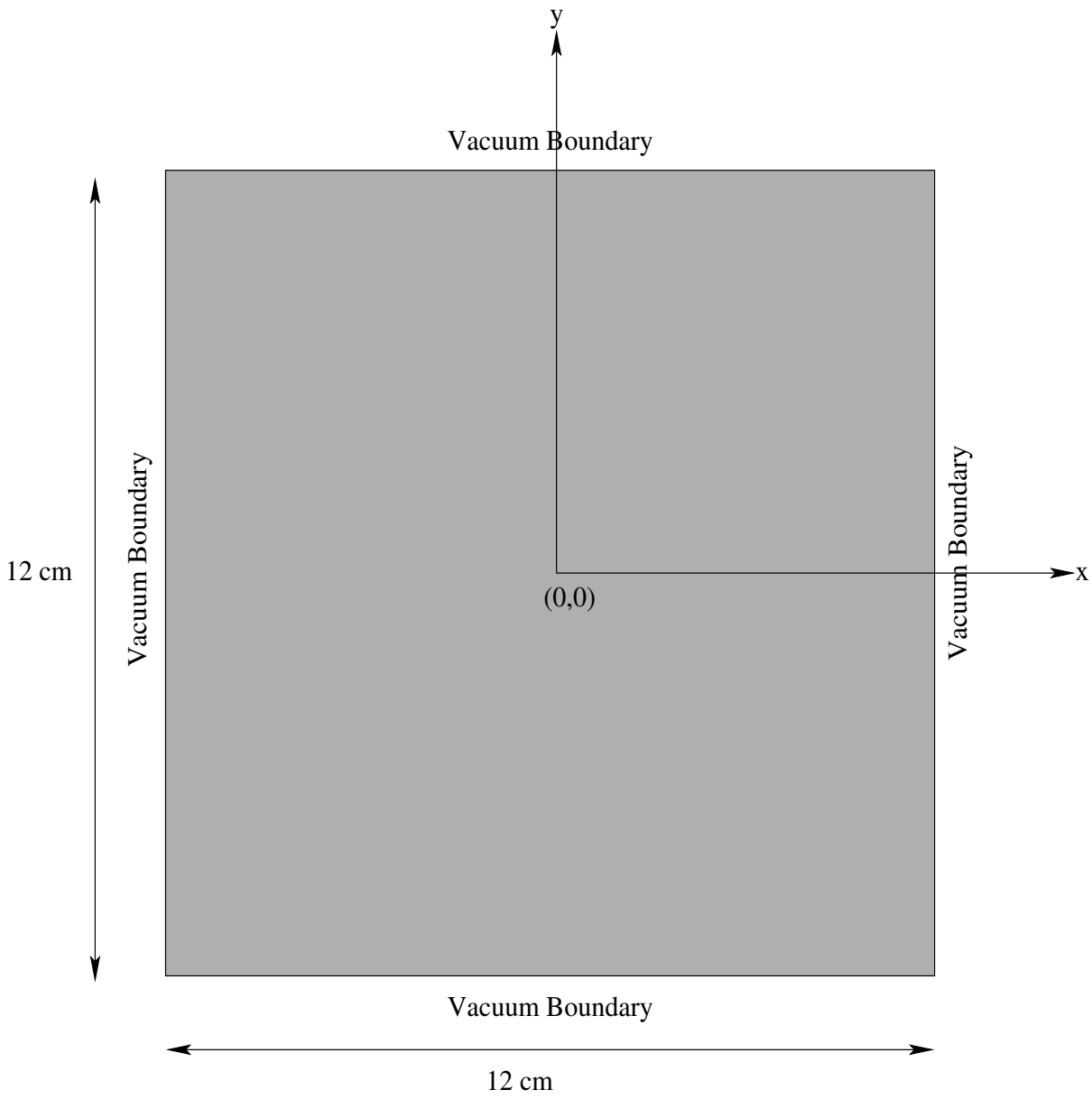
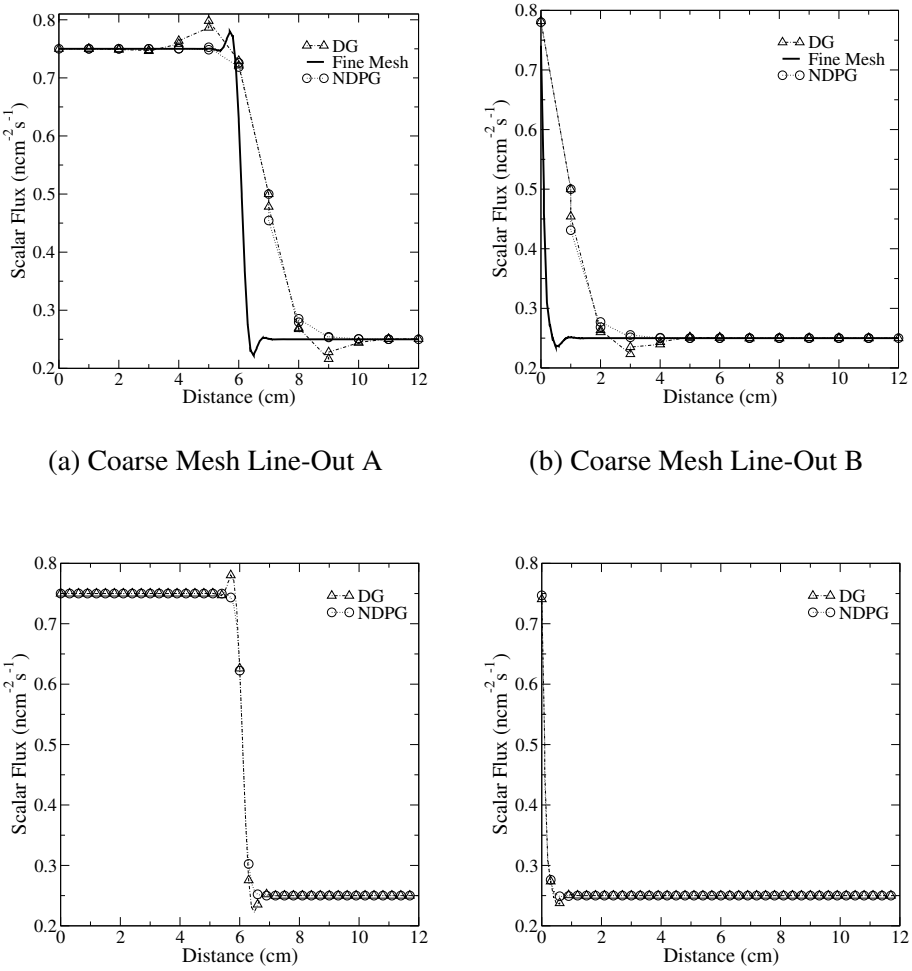


Figure 4.5.: The solution domain of the void problem.



(a) Coarse Mesh Line-Out A

(b) Coarse Mesh Line-Out B

(c) Fine Mesh Line-Out A

(d) Fine Mesh Line-Out B

Figure 4.6.: Scalar flux line-outs through the void problem.

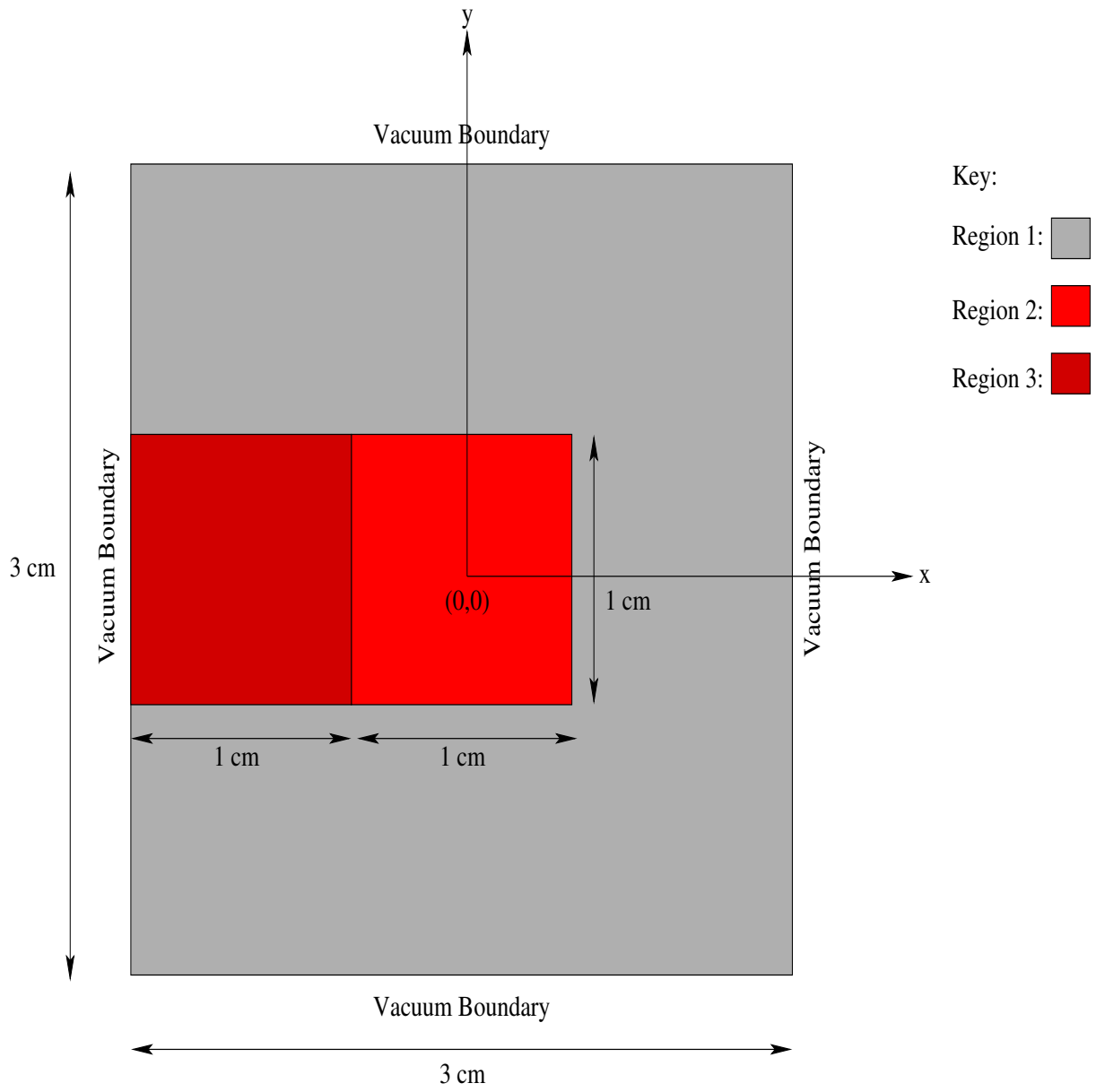
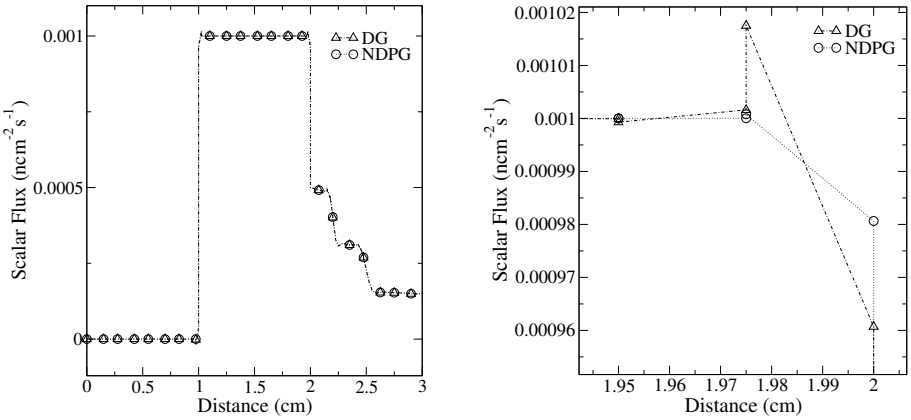


Figure 4.7.: The heavy absorber problem.

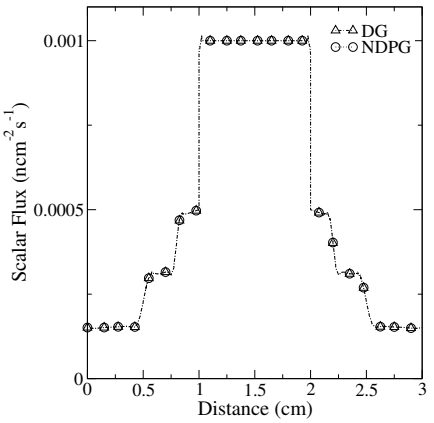
Material Region	Description	Source (n s^{-1})	σ_t (cm^{-1})	σ_s (cm^{-1})
1	Ambient	0.0	0.1	0.0
2	Source	1.0	1000.0	0.0
3	Absorber	0.0	1000.0	0.0

Table 4.2.: Definition of materials in the heavy absorber problem.

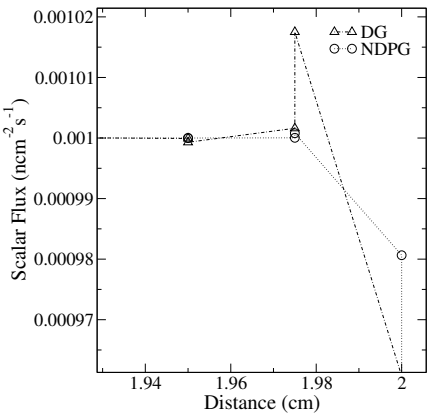


(a) Line-Out A

(b) Line-Out B



(c) Line-Out C



(d) Line-Out D

Figure 4.8.: Scalar flux line-outs through the heavy absorber.

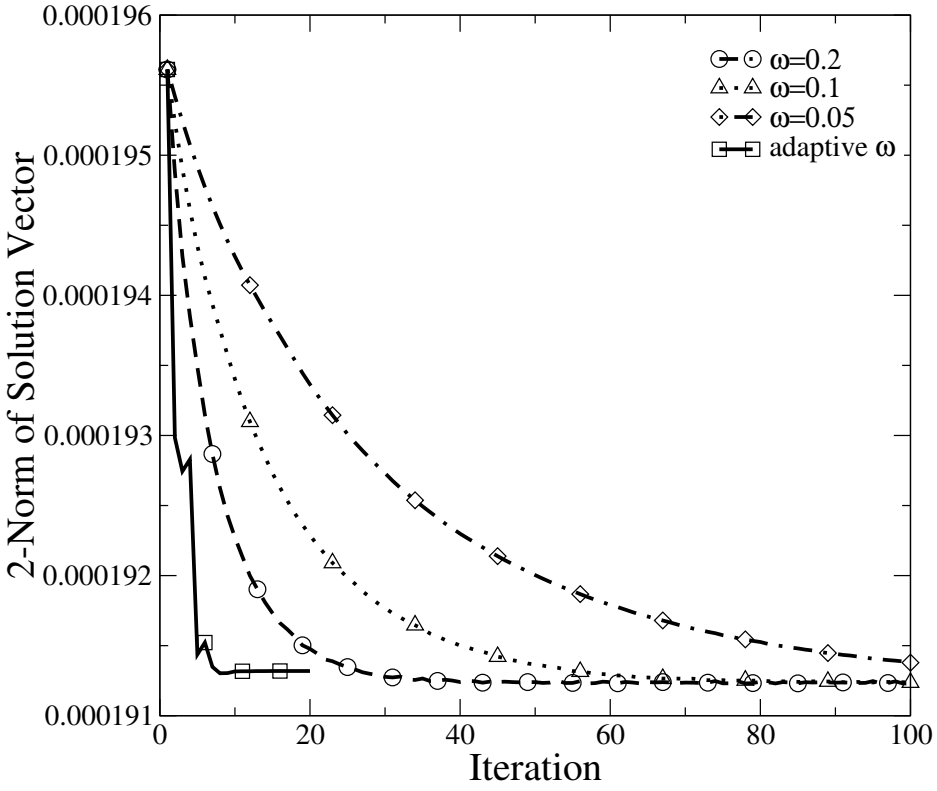


Figure 4.9.: Effect of different relaxation parameters on convergence of the optimal non-linear matrix iteration. Adaptive relax reduces the relaxation parameter ω by a factor 2 each time the 2-norm increases, rapidly accelerating convergence of the matrix solve.

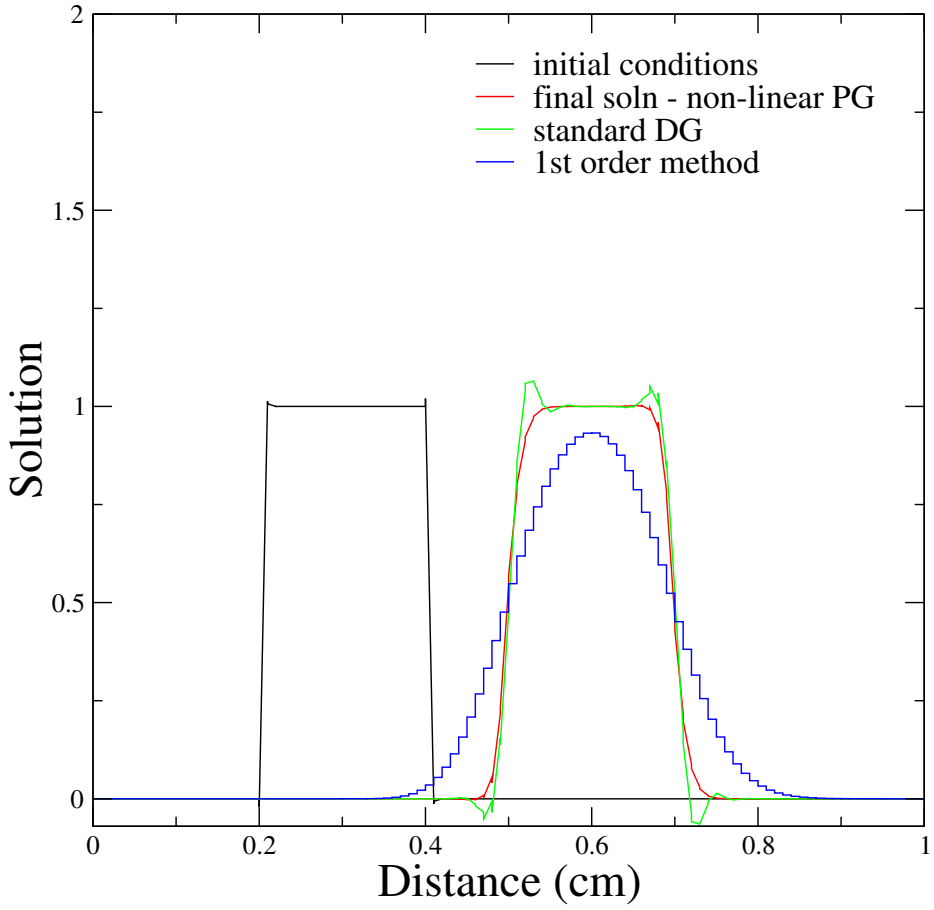


Figure 4.10.: Demonstration of non-linear advection with 100 elements and small time step size.

CHAPTER 5

FUNCTIONAL CORRECTION

Synopsis

This chapter develops an adjoint-based *a posteriori* error method designed to improve computational estimates of a functional. This complements the work of the previous chapters, offering another approach to improving the fidelity of a calculation. Examples of a functional occur in many areas of physics and engineering; they include lift and drag past an obstacle in aircraft and ship design and glacier movement in geology. In nuclear physics they include quantities such as power in a reactor fuel pin, the K_{eff} eigenvalues in a criticality problem or radiation input to a shield or the response of a detector. It is possible to use the adjoint equations to derive an error measure for improving these types of functional. This allows a reliable approximation to the error contained within the functional to be computed. The error may then be subtracted from the functional to improve it. This goal-based scheme also has potential use in automating mesh adaption and is well aligned, therefore, with current work of other groups. The scheme developed in this chapter offers an alternative to adapting the mesh in order to improve a functional.

Contents

5.1. Introduction

In this chapter an adjoint (or sensitivity) based error estimate is formulated which measures the defect contribution of the solution variables to a bulk (volume integrated) functional in source-detector calculations. Bulk functionals can represent a wide range of parameters, such as the K_{eff} eigenvalue in nuclear criticality assessments, or lift and drag past an interface in flow problems such as aerodynamics simulations. Obtaining an estimate of the error within these quantities allows the required accuracy to be attained with significant reductions in computational cost. The error estimate may alternatively be used as a bound to assess reliability of a numerical scheme. The *a posteriori* error measure developed in the current work involves the solution of a primal and its associated adjoint problem. Both solutions are computed in-line, with the adjoint variables used to drive a defect iteration that improves fidelity of the functional integral. This allows the estimate of the functional to be improved and it is shown that the algebraic convergence with increased numerical resolution is improved dramatically with this new method. The improved functional is found to be super-convergent, due to the error within it diminishing faster than the error in the underlying discretisation scheme. In the current work, this is demonstrated using linear discontinuous finite elements on the spatial domain and discrete ordinates and spherical harmonics in angle. In addition to improving functional convergence, the *a posteriori* approach provides error norms with which ultimately automatic mesh adaptivity methods can be applied. The method also indicates where in functional phase space the errors are large. This offers guidance on which phase variable requires adaption. However, even without mesh adaptivity the method indicates which regions of the solution

domain most need to be mesh refined in order to improve functional integrals. The aim of the current work is to improve the accuracy of functional calculations using an adjoint guided defect correction procedure. This scheme may be applied to various bulk functional integrals such as source-detector responses, K_{eff} eigenvalues in criticality assessment (Paraschivoiu and Patera, 1998), and non-fundamental mode calculations (Pierce and Giles, 2000; Muller and Giles, 2001; Power, 2005). Practical applications of this type of method include mesh adaption, with potential use in reaction rate estimations and dosimetry calculations. It is usually the case, in most areas of computational modelling, that the required accuracy can not be attained due to limited computing resources. Adjoint residual methods for obtaining the functional sensitivity to the forward solution at modest resolution have become one of the most viable approaches to assessing functional errors. These errors may be used to improve the accuracy of the functional integral or to assess the reliability of the underlying numerical scheme. Second order defect corrections may be used to provide even sharper error bounds and super-convergent bulk functional estimates, with errors that diminish faster than the errors in the underlying mesh-wise solution on which they depend. Such procedures may be used to indicate where in the bulk functional phase space inaccuracies and other types of sensitivities occur. Not only does this allow assessment of the error, it can be used to guide automatic self adaptivity methods on the grid (Venditti and Darfomal, 2002, 2003, 2000). Thus they have huge implications for improving the fidelity and practicality of large scale problems. Functional calculations in advective radiation transport are generally very expensive computationally, requiring a large number of matrix iterations. This is particularly true of K_{eff} calculations that typically contain a non-linear power iteration on the fission source. Such problems involve the solution of a number of primal equations to achieve a good convergence, for example through Krylov (Arnoldi) subspace iterations. The cost of the adjoint calculation is relatively minor since it is performed only once after the primal solve has converged, and can greatly (as shown here) improve the accuracy of the forward calculation. This is potentially more efficient than using a direct solve on a fine computational grid, and can also be used to determine if any increase in resolution is required. The method contained in the present chapter can be extended to include a combination of defect and adjoint correction methods which have been shown to massively increase the order of accuracy and mesh convergence rate of bulk functional integrals Pierce and Giles (2004). Since the phase-space for functional calculations in radiation transport problems is large, comprising Euclidean

space, angle of neutron travel and energy of the neutrons - six dimensions in total (excluding time), it can be rather unclear how to distribute numerical resolution to achieve accurate results. The method presented provides a systematic way of achieving this. It may also be used to indicate which angles of neutron travel resolution can be eliminated from the discretisation. In multi-group calculations, the method may be used to indicate which energy groups could be eliminated. Rigorous optimisation of neutron energy group structure can be complex (et al, 2005), and the current method may provide a simpler alternative. The resulting energy group structure, angular resolution and spatial discretisation would improve the accuracy of the simulations and the multi-group cross-sections. The advantages of this may also carry over to steady state calculations in which the shape of the fundamental mode can be quite similar to the shape of the flux distributions, for example slow transients in nuclear reactors. Engineering calculations also typically require an estimate of the accuracy of the calculation. The confidence or uncertainty in a prediction can be nearly as important as the bulk quantity itself. Work by Ackroyd showed that K_{eff} eigenvalues converge from above and below as the angular and spatial resolutions are increased for the even-parity and odd-parity principles respectively (Ackroyd, 1997). The method described here has not been applied to multi-group problems, but it has no problem in the formulation for multi-group systems of equations. In the current work, functionals that represent a source-detector response are considered. Linear sensitivities are typical in source-detector type problems, and so emphasis is given to first-order functionals. Higher order functionals represent non-linear sensitivities and are more relevant to processes such as data assimilation. These are beyond the scope of the current work.

The remainder of this chapter has been arranged as follows. Section (5.2) presents the derivation of the equations for the functional sensitivity, defines the functionals used in the present work and derives the residual errors on which the method is based. Examples of a typical functional are defined in section (5.3). In section (5.4) the continuum equations for the error in these functionals is derived, in a manner that is computable. This requires an approximation to a high-order solution to be inserted into the defect equations. A method for obtaining this is described in section (5.5). This leads directly to a computable approximation to the error in the functional, which can be subsequently removed from the initial functional estimate. Numerical results and some analysis are presented in section (5.6) which includes remarks on the numerical efficiency of the approach. Finally,

conclusions are drawn in section (5.7). Recommendations for further work include applying the scheme to K_{eff} eigenvalues.

5.2. Sensitivity Analysis

In this section an error measure is derived based on the primal and adjoint solutions. Suppose the differential equation to be solved may be written in the operator form:

$$\mathscr{L}\mathbf{\Psi}_{exact} - \mathbf{s} = \mathbf{0}, \tag{5.1}$$

for angular discretised linear operator $\mathscr{L}$. In the current work, this operator is defined from the first-order Boltzmann Transport Equation (BTE) to be $\mathscr{L} = \mathbf{A}\cdot\nabla + \mathbf{H}$ where $\mathbf{A}$ is an $\mathscr{M}\times\mathscr{M}$ matrix that contains the Jacobian of the angular discretisation, where $\mathscr{M}$ is the number of moments in the angular expansion. $\mathscr{M}$ depends on the choice of basis function in direction of neutron travel, and on the order of approximation made. For example, for discrete ordinates $\mathscr{M} = n(n+2)$ where n is the S_n order. For spherical harmonics, $\mathscr{M} = \frac{(n+1)(n+2)}{2}$ where n is the order of the P_n expansion. In this definition of $\mathscr{L}$, $\mathbf{H}$ is an $\mathscr{M}\times\mathscr{M}$ matrix containing the scattering-removal data. The extension to non-linear operators is relatively straightforward, although it can involve considerable algebra. $\mathbf{s}$ is a vector of length $\mathscr{M}$ containing the moments of the source and $\mathbf{\Psi}_{exact} = \mathbf{\Psi}_{exact}(\mathbf{r})$ is the angular discrete exact solution, which is also a vector of length $\mathscr{M}$ where $\mathbf{r}$ is the coordinate that remains to be discretised. The vector $\mathbf{0}$ is of length $\mathscr{M}$ and contains zeroes. In a finite element approximation, the solution is represented numerically as $\mathbf{\Psi}(\mathbf{r}) = \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j(\mathbf{r})\mathbf{\Phi_j}$ where $\mathbf{\Phi_j}$ is the computational solution. This is a vector of length $\mathscr{M}$ that contains the solution at node j in the finite element mesh for all the moments, i.e. $\mathbf{\Phi_j} = (\phi_{j_1}, \phi_{j_2}, ..., \phi_{j_{\mathscr{M}}})^T$ where ϕ_{j_μ} is moment μ of the computational solution at node j. $\mathbf{N}_j(\mathbf{r})$ is a diagonal $\mathscr{M}\times\mathscr{M}$ matrix containing the node j basis function. $\mathcal{N}$ is the number of nodes in the finite element mesh. The matrix $\mathbf{N}$ therefore operates on the computational solution vector $\mathbf{\Phi_j}$ at node j for all the moments. The residual of equation (5.1) is given by:

$$\mathscr{L}\mathbf{\Psi}(\mathbf{r}) - \mathbf{s} = \mathscr{R}\left(\mathbf{\Psi}(\mathbf{r})\right). \tag{5.2}$$

From this point onwards, the dependence of $\mathbf{\Psi}$ on the coordinate $\mathbf{r}$ shall be assumed and $\mathbf{\Psi}(\mathbf{r})$ will be written as $\mathbf{\Psi}$. The dependence of $\mathbf{N}_j$ on $\mathbf{r}$ will also be assumed and $\mathbf{N}_j(\mathbf{r})$ will be written as $\mathbf{N}_j$. The aim is to make the residual in

equation (5.2) small in some sense. This is done by multiplying equation (5.2) by a weighting function. In the Discontinuous Galerkin (DG) method this weighting function is chosen to be the basis function $\mathbf{N}_i$, for $i = 1, \mathcal{N}$. Whatever method is chosen to discretise the equations, a matrix equation and residual vector result:

$$\mathbf{r}(\mathbf{\Phi}) = \mathbf{A}\mathbf{\Phi} - \mathbf{S} = \mathbf{0}, \tag{5.3}$$

for $\mathcal{L} \times \mathcal{L}$ matrix $\mathbf{A}$ and discretised source $\mathbf{S}$, where $\mathcal{L} = \mathscr{M} \times \mathcal{N}$ and $\mathbf{S}$ is a vector of length $\mathcal{L}$. The vector $\mathbf{\Phi}$ contains all the vectors $\mathbf{\Phi_j}$ for $j = 1, 2, ..., \mathcal{N}$. i.e. all computational solution vectors on the finite element mesh. However, we note that in the analysis below it is assumed that $\mathbf{r}(\mathbf{\Phi})$ is in some sense a discretised representation of the residual $\mathscr{R}(\mathbf{\Psi})$ multiplied by a representative volume of each cell or node, as achieved in typical finite element Petrov-Galerkin (PG) method or control volume method.

5.3. The Functional or Goal

Suppose that the functional whose accuracy is to be optimised is a function of the angular discrete flux $\mathbf{\Psi}$:

$$F = F(\mathbf{\Psi}). \tag{5.4}$$

Examples of such functionals F on a domain V are:

$$F(\mathbf{\Psi}) = \frac{1}{2}\int_V dV \mathbf{W}\mathbf{\Psi}^2, \quad F(\mathbf{\Psi}) = \int_V dV \mathbf{W}\mathbf{\Psi}, \tag{5.5}$$

and, more generally:

$$F(\mathbf{\Psi}) = \int_V dV f(\mathbf{\Psi}), \tag{5.6}$$

so for the examples given in equation (5.5):

$$f(\mathbf{\Psi}) = \frac{1}{2}\mathbf{W}\mathbf{\Psi}^2 \text{ and } f(\mathbf{\Psi}) = \mathbf{W}\mathbf{\Psi}. \tag{5.7}$$

In these integrals, V is the solution domain and the weighting function $\mathbf{W} = \mathbf{W}(\mathbf{r}) = (W_1(\mathbf{r}), W_2(\mathbf{r}), ..., W_{\mathscr{M}}(\mathbf{r}))^T$ is an angular discrete weighting function in the above, in which $W_\mu(\mathbf{r})$ is the weight function corresponding to moment μ in the angular expansion. For discrete ordinates approximations, W_μ contains the

weight on the chosen quadrature set for angle μ. For spherical harmonics, $W_\mu = 1.0$ for $\mu = 1$ and $W_\mu = 0.0$ for $\mu \neq 1$. Notice that surface integrals may also be written in the form of equation (5.5) using a Dirac delta function on the surface of the domain. This is important in continuous approaches if minimizing a surface integral quantity such as the drag past an object in a moving fluid. In the case of source detector problems, one can restrict the weighting so it is only non-zero in the body of the detector, and so W_μ will generally depend on $\mathbf{r}$. For example, in equation (5.5) the weighting function $\mathbf{W}$ might be unity in the local vicinity of an observation point of interest in the solution domain and zero elsewhere, and in general this will be different for different moments. However, $F(\mathbf{\Psi})$ may be any derived quantity of the flux $\mathbf{\Psi}$. Applying a first-order Taylor series, the gradient $\frac{\partial f}{\partial \mathbf{\Psi}}$ near the angular discrete exact solution $\mathbf{\Psi}_{exact}$ can be obtained from:

$$\left(\frac{\partial f}{\partial \mathbf{\Psi}}\right)^T (\mathbf{\Psi}_{exact} - \mathbf{\Psi}) \approx f(\mathbf{\Psi}_{exact}) - f(\mathbf{\Psi}), \tag{5.8}$$

or in discrete form:

$$\left(\frac{\partial F}{\partial \mathbf{\Psi}}\right)^T (\mathbf{\Psi}_{exact} - \mathbf{\Psi}) \approx F(\tilde{\mathbf{\Psi}}_{exact}) - F(\mathbf{\Psi}), \tag{5.9}$$

in which $\mathbf{\Psi}_{exact}$ is a vector containing the exact solution at the $\mathcal{N}$ finite element nodes (or control volume cells) and $\tilde{\mathbf{\Psi}}_{exact} = \sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_{exact_j}$.

5.4. Continuum Error Measure

Similarly, the continuum residual may be expanded in a first-order Taylor series:

$$\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^T (\mathbf{\Psi}_{exact} - \mathbf{\Psi}) \approx \mathscr{R}(\mathbf{\Psi}_{exact}) - \mathscr{R}(\mathbf{\Psi}), \tag{5.10}$$

where $\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)$ is a vector containing the derivative of the residual $\mathscr{R}$ with respect to the flux $\mathbf{\Psi}$. Since $\mathscr{R}(\mathbf{\Psi}_{exact}) = \mathbf{0}$:

$$(\mathbf{\Psi}_{exact} - \mathbf{\Psi}) \approx -\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^{-T} \mathscr{R}(\mathbf{\Psi}), \tag{5.11}$$

where row μ of the vector $\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^{-T}$ contains the inverse of the row μ component of $\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^{T}$. Equation (5.11) is then combined with equation (5.8) to obtain:

$$f(\mathbf{\Psi}_{exact}) - f(\mathbf{\Psi}) \approx -\left(\frac{\partial f}{\partial \mathbf{\Psi}}\right)^{T}\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^{-1}\mathscr{R}(\mathbf{\Psi}). \tag{5.12}$$

Integrating equation (5.12) over the domain V and using Green's theorem and ignoring the resulting surface integrals results in:

$$F(\mathbf{\Psi}_{exact}) - F(\mathbf{\Psi}) \approx -\int_{V} dV \left(\frac{\partial f}{\partial \mathbf{\Psi}}\right)^{T}\left(\frac{\partial \mathscr{R}}{\partial \mathbf{\Psi}}\right)^{-1}\mathscr{R}(\mathbf{\Psi})$$

$$= -\int_{V} dV \left(\mathscr{R}(\mathbf{\Psi})\right)^{T}\left(\frac{\partial \mathscr{R}^{*}}{\partial \mathbf{\Psi}^{*}}\right)^{-1}\frac{\partial f}{\partial \mathbf{\Psi}} = -\int_{V} dV \left(\mathscr{R}(\mathbf{\Psi})\right)^{T}\mathbf{\Psi}^{*}_{exact}, \tag{5.13}$$

in which $\mathscr{R}^{*}(\mathbf{\Psi}^{*}) = \mathscr{L}^{*}\mathbf{\Psi}^{*}$ is the adjoint residual, which is generated by the adjoint operator $\mathscr{L}^{*}$ acting on the adjoint solution $\mathbf{\Psi}^{*}$. The definition of the adjoint operator is used in the above expression, and reorders the terms in the integrand. This definition is derived in appendix (C). The quantity $\mathbf{\Psi}^{*}_{exact}$ is the exact adjoint solution, which satisfies the equation:

$$\left(\frac{\partial \mathscr{R}^{*}}{\partial \mathbf{\Psi}^{*}}\right)(\mathbf{\Psi}^{*}_{exact}) = \frac{\partial f}{\partial \mathbf{\Psi}}. \tag{5.14}$$

Equation (5.14) is derived by first considering the adjoint equation. This may equivalently be written in residual form or operator form:

$$\mathscr{R}^{*} = \mathscr{R}^{*}(\mathbf{\Psi}^{*}) = \mathscr{L}^{*}\mathbf{\Psi}^{*} - \frac{\partial f}{\partial \mathbf{\Psi}}, \tag{5.15}$$

in which $\mathscr{L}^{*}$ is the adjoint operator and $\frac{\partial f}{\partial \mathbf{\Psi}}$ the source term in the adjoint system of equations. One then differentiates this with respect to the adjoint eigenvector. This leaves just the adjoint operator $\mathscr{L}^{*}$:

$$\frac{\partial \mathscr{R}^{*}}{\partial \mathbf{\Psi}^{*}} = \frac{\partial}{\partial \mathbf{\Psi}^{*}}\left(\mathscr{L}^{*}\mathbf{\Psi}^{*} - \frac{\partial f}{\mathbf{\Psi}}\right) = \mathscr{L}^{*}, \tag{5.16}$$

which is defined as the derivative of the adjoint residual. Placing this back into equation (5.15) one obtains:

$$\mathscr{R}^{*} = \mathscr{R}^{*}(\mathbf{\Psi}^{*}) = \frac{\partial \mathscr{R}^{*}}{\partial \mathbf{\Psi}^{*}}\mathbf{\Psi}^{*} - \frac{\partial f}{\mathbf{\Psi}}, \tag{5.17}$$

and because $\mathscr{R}^*(\mathbf{\Psi}^*_{exact}) = \mathbf{0}$, one will note from the above expression that:

$$\frac{\partial \mathscr{R}^*}{\partial \mathbf{\Psi}^*}\mathbf{\Psi}^*_{exact} = \frac{\partial f}{\mathbf{\Psi}}, \tag{5.18}$$

which is that used in equation (5.13) above. The exact adjoint solution is treated as an unknown quantity, because it is associated with the fine mesh that is computationally prohibitive. One may proceed by decomposing the right hand side of equation (5.13) in the following manner:

$$\begin{aligned} F(\mathbf{\Psi}_{exact}) - F(\mathbf{\Psi}) \approx &- \int_V dV \left(\mathscr{R}(\mathbf{\Psi})\right)^T \mathbf{\Psi}^*_h \\ &+ \int_V dV \left(\mathscr{R}(\mathbf{\Psi})\right)^T (\mathbf{\Psi}^*_h - \mathbf{\Psi}^*_{exact}), \end{aligned} \tag{5.19}$$

where $\mathbf{\Psi}^*_h$ is the coarse mesh adjoint solution projected onto the fine mesh. A method for obtaining $\mathbf{\Psi}^*_{\frac{h}{2}} \approx \mathbf{\Psi}^*_{exact}$ (the exact solution is unknown) is discussed in section (5.5). Typically, $\mathbf{\Psi}^*_h$ and $\mathbf{\Psi}^*_{\frac{h}{2}}$ are solved for using a linear and a higher-order finite element method, in which $\mathbf{\Psi}^* = \mathbf{\Psi}^*(\mathbf{r}) = \sum_{j=1}^{\mathcal{N}} \mathbf{N}(\mathbf{r})_j \mathbf{\Phi}^*_j$ where $\mathbf{\Phi}^*_j = (\phi^*_{j1}, \phi^*_{j2}, ..., \phi^*_{j\mathscr{M}})^T$ is a vector of length $\mathscr{M}$ containing the adjoint solution for each moment at node j. The first term on the right of equation (5.19) is therefore computable, and maybe applied as a correction. The second term in equation (5.19) contains the error in this computable correction. However, in the case of Galerkin projections the first term is shown not to contribute:

$$\begin{aligned} \int_V dV \left(\mathscr{R}(\mathbf{\Psi})\right)^T \mathbf{\Psi}^*_h &= \int_V dV \left(\mathscr{R}(\mathbf{\Psi})\right)^T \left(\sum_{j=1}^{\mathcal{M}} \mathbf{M}_j \mathbf{\Phi}^*_{h_j}\right) \\ &= \sum_{j=1}^{\mathcal{M}} \mathbf{\Phi}^*_{h_j} \int_V dV \mathbf{M}_j \mathscr{R}(\mathbf{\Psi}) = \mathbf{0}. \end{aligned} \tag{5.20}$$

In such cases, the error is approximated by the second term on the right hand side of equation (5.19) which is used as a correction. Interestingly, it is found that:

$$\int_V dV \left(\mathscr{R}(\mathbf{\Psi})\right)^T (\mathbf{\Psi}^* - \mathbf{\Psi}^*_h) = \int_V dV \left(\mathscr{R}^*(\mathbf{\Psi}^*)\right)^T (\mathbf{\Psi} - \mathbf{\Psi}_h), \tag{5.21}$$

which maybe used as an approximation to the second term on the right hand side of equation (5.19), which can now be used directly to determine an improved pre-

diction of F, that is to obtain an estimate of $F(\mathbf{\Psi}_{exact})$. This will contain a smaller error than the estimate of $F(\mathbf{\Psi})$, provided $\mathbf{\Psi}_{exact}$ can be approximated.

5.5. High-Order Solution

In this section the method for obtaining $\mathbf{\Psi}_h$ used in the previous section, and in equation (5.21) is described. This high-order solution is obtained via a post-processing of the low-order (coarse-mesh) solution. The sensitivity in the angular flux solution is represented by $\mathbf{\Psi}_h - \mathbf{\Psi}$, in which $\mathbf{\Psi}$ is the finite element solution obtained from a low order approximation (for example linear finite elements) and $\mathbf{\Psi}_h$ is that obtained from a higher order approximation, for example quadratic finite elements that use the linear solution as the incoming boundary condition. This is found to be the most accurate way to interpolate between the points on the linear solution. The purpose of using a quadratic element is to obtain a solution $\mathbf{\Psi}_h$ that is differentiable more times, and therefore smoother, than $\mathbf{\Psi}$. Thus the quantity $\mathbf{\Psi}_h - \mathbf{\Psi}$ may be used as a smooth reconstruction of the angular flux solution inside an element. Essentially, this is to provide an approximation to a continuous solution which is not known. $\mathbf{\Psi}_h$ is obtained using the linear solution $\mathbf{\Psi}$ on the boundary nodes of the quadratic element, and then solving the resulting system of equations associated with the quadratic element. A set of quadratic shape func-

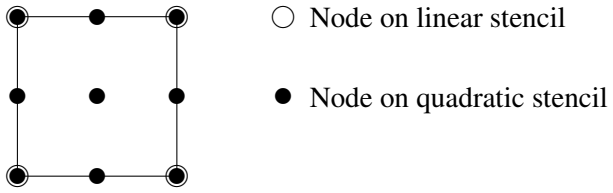


Figure 5.1.: **Smoothing of Linear Solution**

tions $\mathbf{M} = \mathbf{M}(\mathbf{r})$ is chosen, in which $\mathbf{M}_i = \mathbf{M}_i(\mathbf{r})$ is a diagonal $\mathscr{M} \times \mathscr{M}$ matrix containing the quadratic basis function for node i inside the quadratic element, and $\mathscr{M}$ is the number of angular moments in the linear and quadratic solutions. The quadratic solve is done internal to each element, using a DG projection. This is done in the same way as the linear solution is discretised, except using quadratic basis function $\mathbf{M}_i$ instead of linear basis function $\mathbf{N}_j$. That is to use the diagonal $\mathscr{M} \times \mathscr{M}$ matrix $\mathbf{M}_i$ which contains the quadratic basis function for node i in the quadratic mesh. The linear DG solution (known at the points marked on figure

(5.1) with the empty circles) is interpolated on the surface of the element to obtain a full vector of incoming information entering the quadratic element. This is placed in the quadratic surface integral. This approach successfully interpolates between the data points on the low order solutions. $\mathbf{\Phi} = (\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_{\mathcal{N}})^T$ internal to each element. This obtains a smoother solution at the points marked in figure (5.1) by a solid circle. The quadratic solution is approximated numerically with $\mathbf{\Psi}_h = \sum_j^{\mathcal{M}} \mathbf{\Phi}_{h_j}$ in which the vector solved for is $\mathbf{\Phi}_{h_i} = (\phi_{h_{i_1}}, \phi_{h_{i_2}}, ..., \phi_{h_{i_{\mathscr{M}}}})^T$, where $\phi_{h_{i_\mu}}$ is moment μ of the solution at node i and $\mathcal{M}$ is the number nodes in the quadratic mesh. Each component of $\mathbf{\Phi}_h = (\mathbf{\Phi}_{h_1}, \mathbf{\Phi}_{h_2}, ..., \mathbf{\Phi}_{h_{\mathcal{M}}})^T$ is a vector of length $\mathscr{M}$ where $\mathscr{M}$ is the number of moments in the angular approximation. The discretised quadratic equation that contains the linear solution in the inbound surface integral is written:

$$\begin{aligned}
&\int_V dV \left(-\mathbf{A} \cdot \nabla \mathbf{M}_i \left(\sum_{j=1}^{\mathcal{M}} \mathbf{M}_j \mathbf{\Phi}_{h_j} \right) + \mathbf{H} \mathbf{M}_i \left(\sum_{j=1}^{\mathcal{M}} \mathbf{\Phi}_{h_j} \right) \right) \\
&\qquad + \int_{\Gamma_{\mathbf{n}\cdot\mathbf{A}\geq 0}} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{M}_i \left(\sum_{j=1}^{\mathcal{M}} \mathbf{M}_j \mathbf{\Phi}_{h_j} \right) \\
&= \int_V dV \mathbf{M}_i \left(\sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{q_j} \right) - \int_{\Gamma_{\mathbf{n}\cdot\mathbf{A}<0}} d\Gamma \mathbf{n} \cdot \mathbf{A} \mathbf{M}_i \left(\sum_{j=1}^{\mathcal{N}} \mathbf{N}_j \mathbf{\Phi}_j \right). \qquad (5.22)
\end{aligned}$$

The surface terms arise when integration by parts is applied to the advection terms, as done for the linear discretisation. The first surface integral contains outbound information, the flow of which is determined by the condition $\mathbf{n} \cdot \mathbf{A} \geq \mathbf{0}$. This is placed on the left side of the equation and is absorbed into the quadratic solution vector $\mathbf{\Phi}_h$ at the $\mathcal{M}$ nodes in the quadratic stencil. The second surface integral contains the linear solution $\mathbf{\Phi}$ at the $\mathcal{N}$ points in the linear stencil. This is the incoming information the flow of which is determined by the condition $\mathbf{n} \cdot \mathbf{A} < 0$ at the boundary. Therefore the mass matrix, element i, j of which is given by $\int_V \mathbf{M}_i \mathbf{N}_j dV$ where i is the column number and j the row number, will not form a square matrix. Note that $\mathbf{M}_i$ and $\mathbf{N}_j$ are $\mathscr{M} \times \mathscr{M}$ diagonal matrices containing the node i quadratic shape function and the node j linear shape function respectively. Thus the mass matrix will be an $\mathcal{M}$ by $\mathcal{N}$ matrix that maps the linear solution vector $\mathbf{\Phi}_j$ at node $j = 1, 2, ..., \mathcal{N}$ in the linear stencil on to node $i = 1, 2, ..., \mathcal{M}$ in the quadratic stencil, as done in the inbound surface integral in the above equation.

The quantity $\mathbf{q_j}$ in the above equation is a vector of length $\mathscr{M}$ containing the point source at node $j = 1, 2, ..., \mathcal{N}$ in the linear stencil.

5.6. Numerical Examples

In this section results are presented that demonstrate the adjoint-informed defect correction method. A simple linear functional and a second functional that has quadratic dependence on the flux solution are improved by the method, with considerable enhancement of the convergence rate. Therefore a given error may be obtained on a coarser mesh interval when the correction scheme is applied. Even in demanding two dimensional problems with highly localised sources and detectors, the defect correction method performs consistently well. Results are demonstrated in one spatial dimension for both types of functional, and in two spatial dimensions for the linear functional. Correction of the quadratic functional in two spatial dimensions remains the topic of further work.

5.6.1. Correction on One-Dimensional Meshes

A simple test using one spatial dimension is used to investigate the potential of the scheme. The problem comprises a source region in the range $0.0 \leq x \leq 0.25$ and a detector in the range $0.75 \leq x \leq 1.0$, on the domain V with boundary ∂V where $0.0 \leq x \leq 1.0$. This domain contains a single pure absorber of cross-section $\sigma = 5$. Two types of functional are used, the first type a simple linear functional of the form $F = \int_V dx \mathbf{W}(x)\mathbf{\Psi}(x)$ and the second of the form $F = \frac{1}{2}\int_V dx \mathbf{W}(x)\mathbf{\Psi}^2(x)$. The function $\mathbf{W}(x)$ is a weighting factor that restricts the contribution of ψ to the functional to a spatial region of interest. For example, inside a detector region the weighting function may be unity, and outside the detector it may be zero. Both functional forms are calculated internal to the domain V, and so the surface ∂V has no contribution to the error integral. Figure (5.2a) shows a plot of the error in the functional estimate against different meshes. A log space is used so the slopes of the plots provide the convergence rate. Standard linear Discontinuous Galerkin (DG) finite elements are used for the spatial projection. The result one obtains without using any correction is labelled $F - F_{exact}$. It is observed from the slope of this plot that the uncorrected linear functional has a convergence rate of $O(3)$, as expected. When the defect correction is applied, the plot labelled $F_{imp} - F_{exact}$ is obtained. This shows a convergence rate of $O(5)$.

In figure (5.2b) the comparison is repeated for the case of the quadratic functional. This problem is more challenging as it depends on a higher-order variation of the flux, and a slower convergence rate is to be expected of the corrected functional. This is because the error correction scheme, being based on a first-order Taylor series, operates to first order only and is not able to correct for errors in the higher terms[1]. Consequently, such errors remain even after the present scheme has been applied. A rate of $O(3)$ is again observed in the uncorrected quadratic functional. The defect correction method, however, increases the convergence rate to $O(4)$ by recovering some of the error in the functional. Thus there are strong efficiency arguments in favour of the method, since a smaller error is obtained on a coarser grid, than one can obtain with the uncorrected functional.

5.6.2. Correction in Two Dimensions: Problem 1

The functionals may also be corrected on two dimensional meshes. This is shown for $x - y$ Cartesian geometry though extension to other geometries is straightforward. The domain used in this problem is represented by $V = x, y$ and has a boundary ∂V. The problem domain of the first two-dimensional test case is illustrated in figure (5.3), and the materials used in this domain are specified in table (5.1). Results are demonstrated using both discrete ordinates and spherical harmonics angular discretisations. Correction is performed on the linear functional estimate. The source region of strength 1 ns^{-1} is imposed in the range $0.0 \leq x \leq 0.25$, $0.0 \leq y \leq 1.0$ and the detector in the range $0.75 \leq x \leq 1.0$, $0.0 \leq y \leq 1.0$. A pure absorber is used for which cross-section $\sigma = 5$. Vacuum boundary conditions are used on ∂V. Results are included in figure (5.4) for (a) a P_3 angular approximation and (b) an S_6 angular approximation. For each angular representation, the functional error is plotted in log space against mesh size. This allows the slopes, and hence the convergence rate to be determined. It is found that in the P_3 case, the uncorrected functional error has a slope of 2.94. Applying the defect correction increases this to 3.90. It is noted that convergence is slightly poorer in multi-dimensions than in one-dimensional cases, due to the reduced order of the couple between the elements. This arises due to the surface terms in the discretisation and is characteristic of the DG method. It is expected, though not shown in the present work, that in three-dimensions the convergence rate will be no worse than it is in two-dimensions. In the case of the S_6 calculation, the uncor-

[1] this has implications for K-eigenvalues which is addressed in chapters (6) and (7).

rected functional estimate shows a slope of 2.90 also. Applying the correction in this case increases the slope to 3.60. It is expected the ray-effect may in general have an adverse effect on the degree of correction obtainable. In P_n calculations, the smoothness of the solution results in a more uniform distribution of the error throughout the domain.

5.6.3. Correction in Two Dimensions: Problem 2

The problem domain of the second two-dimensional test case is illustrated in figure (5.5), and the materials used in this domain are specified in table (5.2). This problem is more demanding than Problem 1, due to the highly localised source and detector regions. Results are again demonstrated using both discrete ordinates and spherical harmonics in direction of neutron travel. This demonstrates the error correction is not sensitive to the choice of angular basis function. This is a desirable property for the error correction method to have, allowing it to be deployed within different angular frameworks. The source of strength 1 ns^{-1} is imposed in the range $0.0 \leq x \leq 0.25$, $0.0 \leq y \leq 0.25$. The detector is located in the region $0.75 \leq x \leq 1.0$, $0.75 \leq y \leq 1.0$. As in Problem, vacuum boundaries are used on ∂V and a pure absorber of cross-section $\sigma = 3$ specified as shown in table (5.2). Results are included for the linear functional in figure (5.6). In this case, two different angular resolutions are considered to investigate how detrimental low order approximations are to this type of defect correction. Figure (5.6a) shows a P_1 correction. The uncorrected functional error has a slope of 3.11. Correction increases this to 3.61. Using an S_2 angular representation, shown in figure (5.6b) the slope is increased from 1.89 (uncorrected) to 2.27 (corrected). Increasing the order of the P_n approximation to P_5, slopes of 2.97 (uncorrected) and 3.89 (corrected) are obtained. Increasing the order of the S_n calculation to S_8, the slope increases from 2.34 (uncorrected) to 2.83 (corrected). It is seen that convergence rate is improving for the S_n method as order of angular approximation is increased. This may be due to smoothing of the solution.

5.6.4. Numerical Efficiency

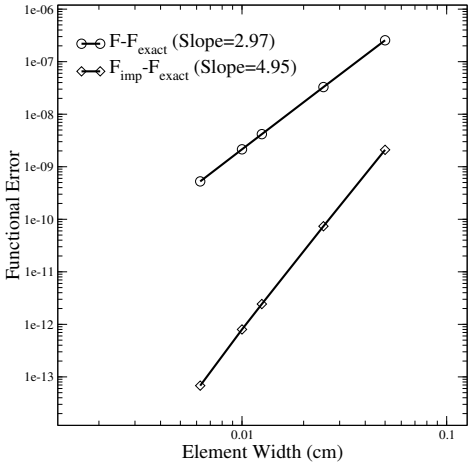
An important consideration in the current work is efficiency of the numerical scheme. The objective is to obtain high fidelity solutions on coarse meshes, whose native accuracy is unacceptably poor. Fine-mesh solutions have the desired accuracy, however directly obtaining them is computationally prohibitive. A practical

approach is to obtain the functional on a coarse mesh and improve the convergence with a defect iteration. This is done in the present work using adjoint-informed recovery of the bulk error the coarse mesh result contains. The error recovery scheme involves considerable computational overhead. The procedure is to first obtain the coarse mesh result using a Riemann-based linear finite element method. Typically, this requires many matrix iterations to achieve convergence. Once the linear solution has been obtained on the coarse mesh, only one further iteration is performed to project it onto the high resolution stencil. This projection operation, although considerably more intensive than one linear solution iteration, comprises a fraction the cost of the combined overhead of all the linear solution iterations. It represents a very small addition to the forward solution overhead. This is because only one matrix iteration is required to project the converged linear result. The linear solution iteration is then repeated for the adjoint solution. This is approximately the same overhead as the forward linear solution. Therefore, the total overhead of the corrector scheme is slightly more than double that of a basic linear solve without any defect correction. To obtain the high-order solution directly (which one can expect to contain a similar error to a corrected low-order solution), requires significantly greater overhead than this. In the case of problem 1, it is noted from figure (5.4) that approximately 8 times the number of elements are needed in a direct solve to reduce the bulk functional error to the order seen in the corrected result. There are thus strong efficiency arguments in favour of this type of defect correction scheme.

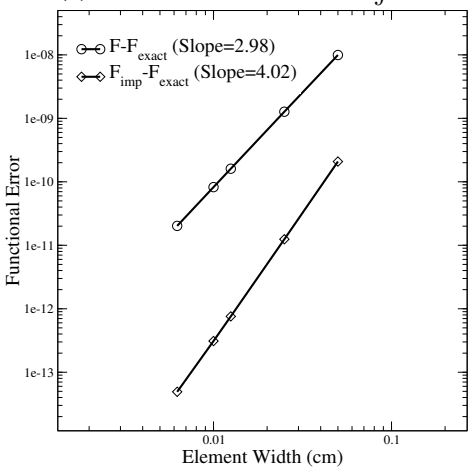
5.7. Conclusions

In this chapter a sensitivity or goal based error norm is defined for bulk functional calculations in source detector problems in neutron transport. It is shown how this can be applied to different types of functional using linear discontinuous finite elements. The advantage of this approach are three fold: the super-convergence of the goal with future mesh adaptivity and indication of where resolution is required in manual meshing/angle refinement; the ability to assign bounds on the accuracy of this goal; and the possibility of improving the accuracy of the goal. The application to relatively simple functional problems demonstrates the robustness of this approach. A second order correction scheme maybe combined with the current approach, and this is expected to provide even sharper error bounds and improve the defect correction for more complicated (non-linear) functionals. Chapters (6)

and (7) concentrate on using this method for K_{eff} eigenvalue defect correction. It may also be used to guide automatic self-adaptivity methods in which the indicated need for resolution in space, angle and energy will be exploited. However, mesh adaptivity is currently being pursued by other workers (Baker, 2011) and has not been considered in this Thesis.



(a) Linear Functional $F = \int dVW\Psi$



(b) Quadratic Functional $F = \frac{1}{2} \int dVW\Psi^2$

Figure 5.2.: **Functional Correction in 1-Dimension**

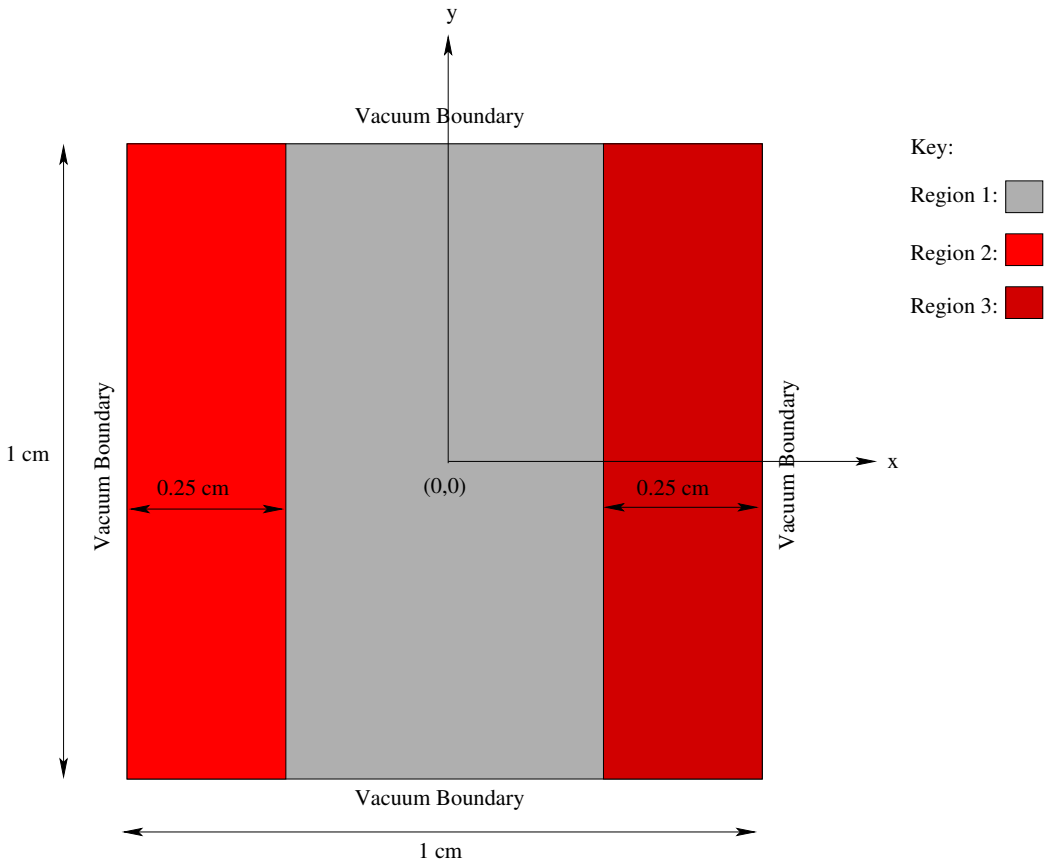
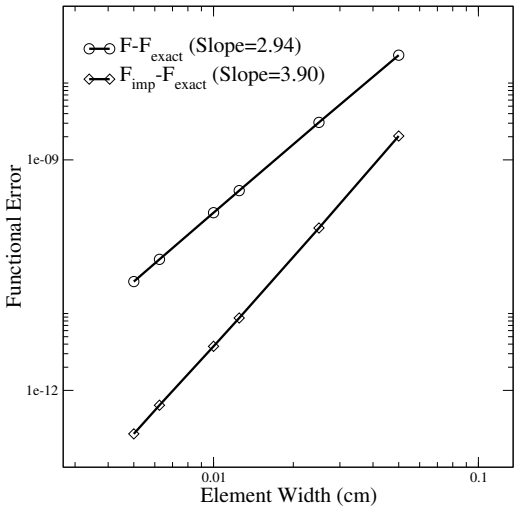


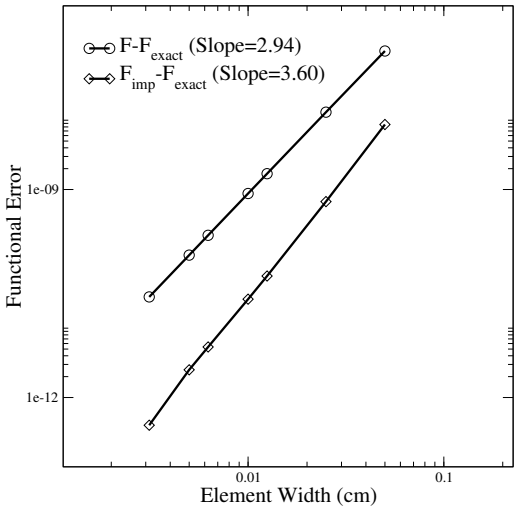
Figure 5.3.: **Problem 1 Domain**

Material Region	Description	Source (n s^{-1})	σ_t (cm^{-1})	σ_s (cm^{-1})
1	Ambient	0.0	5.0	0.0
2	Source	1.0	5.0	0.0
3	Detector	0.0	5.0	0.0

Table 5.1.: Definition of materials in problem 1.



(a) P_3 Angular Approximation



(b) S_6 Angular Approximation

Figure 5.4.: **Linear Functional Correction in Problem 1**

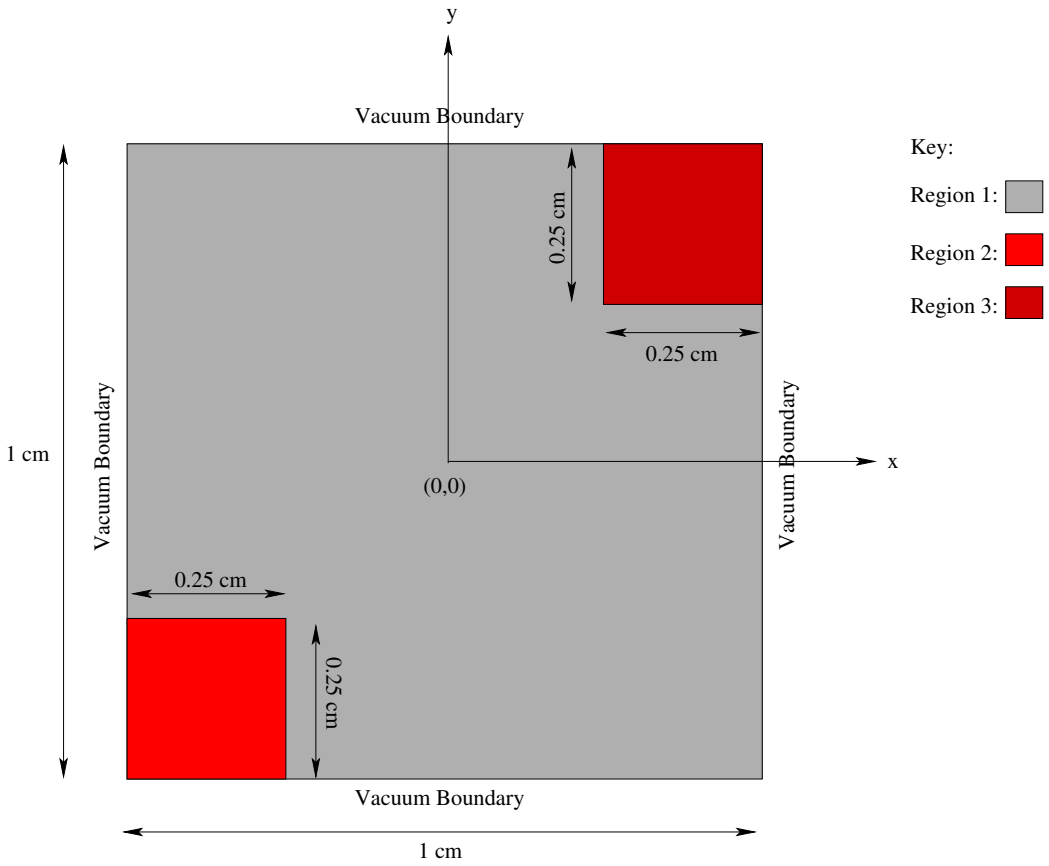
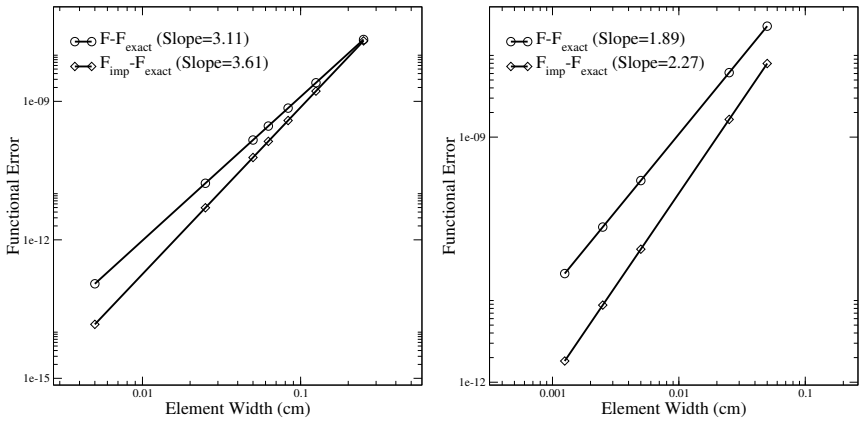


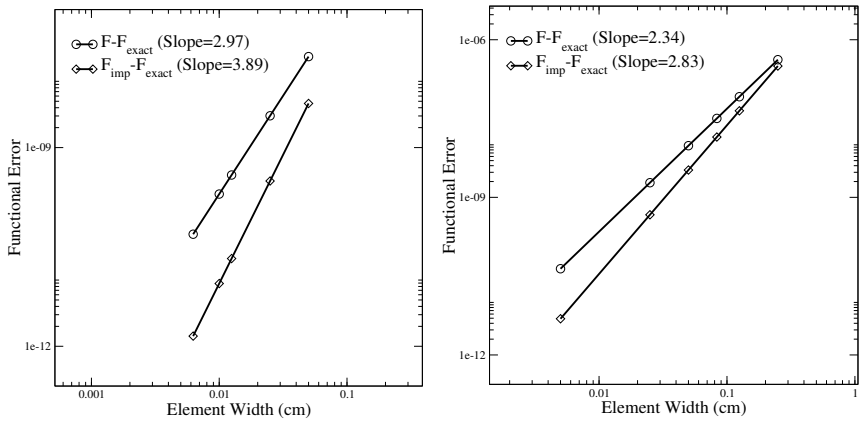
Figure 5.5.: **Problem 2 Domain**

Material Region	Description	Source (n s^{-1})	σ_t (cm^{-1})	σ_s (cm^{-1})
1	Ambient	0.0	3.0	0.0
2	Source	1.0	3.0	0.0
3	Detector	0.0	3.0	0.0

Table 5.2.: Definition of materials in problem 2.



(a) P_1 Angular Approximation (b) S_2 Angular Approximation



(c) P_5 Angular Approximation (d) S_8 Angular Approximation

Figure 5.6.: **Linear Functional Correction in Problem 2**

CHAPTER 6

EIGENVALUE CORRECTION IN ELLIPTIC PROBLEMS

Synopsis

The adjoint-based *a posteriori* error measure developed in the previous chapter is applied in the present chapter to eigenvalue problems. This is demonstrated for elliptic operators in diffusive problems. The K_{eff} eigenvalue is first obtained using a conventional inverse power iteration on the fission source, for the forward system of equations using a simple linear finite element type. The solution procedure is then repeated using the adjoint equations. The eigenvector solution to the adjoint system is enriched in a post-processor step, and convolved with the residual of the forward equations. This produces a reliable approximation to the error in the eigenvalue. This approximation to the error is then subtracted from the eigenvalue producing a better estimate. It is shown how this approach can accelerate the mesh convergence of the eigenvalue in smooth, diffusive problems. The chapter concludes by recommending an approach be outlined for achieving this with hyperbolic operators in transport problems.

Contents

6.1. Introduction

A scheme for improving the accuracy and reducing the error in eigenvalue calculations is presented. Using a first order Taylor series expansion of both the eigenvalue solution and the residual of the governing equation, an approximation to the error in the eigenvalue is derived. This is done using a convolution of the equation residual and adjoint solution, which is calculated in-line with the primal solution. A defect iteration on the solution is then performed in which the approximation to the error is used to apply a correction to the eigenvalue. The method is shown to dramatically improve convergence of the eigenvalue. The equation for the eigenvalue is shown to simplify when certain normalisations are applied to the eigenvector. Two such normalisations are considered; the first of these is a fission-source type of normalisation and the second is an eigenvector normalisation. Results are demonstrated on a number of demanding elliptic problems using continuous Galerkin weighted finite elements. Moreover, the correction scheme may also be applied to hyperbolic problems and arbitrary discretisation. This is not limited to spatial corrections and may be used throughout the phase space of the discrete equation. The applied correction not only improves fidelity of the calculation, it allows assessment of the reliability of numerical schemes to be made and could be used to guide mesh adaption algorithms or to automate mesh generation schemes.

Eigenvalue problems arise in many areas of science, mathematics and engineering. They characterise a diverse range of systems that are of interest such as glacier movements in geology, to lift and drag past obstructions to flow in aircraft and

ship design. In mathematics they describe the orthogonal properties of a matrix and in reactor physics the distribution of neutrons in a lattice, allowing criticality to be assessed. Accurate determination of the system eigenvalue is clearly important in a wide range of applications. Computational procedures for obtaining the eigenvalue are typically very intensive numerically, notably so in criticality problems where a large number of non-linear iterations are required to accurately characterise the system. Coarsened computational meshes make calculations of the eigenvalue numerically feasible, however they are unable to capture or contain enough information about the problem to achieve solutions of acceptable accuracy. Finer meshes that offer reasonable accuracy are in many cases unfeasible where highly iterative solution schemes are employed, such as source iteration schemes used to obtain eigenvalues in neutron transport applications. This has motivated the development of methods that improve the numerical solution on meshes that would otherwise not offer sufficient accuracy. An example of this type of approach is the *a posteriori* error measure, in which the solution itself is used in some way in order to obtain a defect estimate Ainsworth and Oden (1997). Another popular approach is to use the adjoint problem to obtain an approximation to the error, and subsequently remove this approximation from the solution obtaining improved functional estimates Venditti and Darfomal (2000); Pierce and Giles (2004); Giles et al. (2004). Alternatively, one might use it as a metric to guide a grid adaption step. This has been shown to be successful in Venditti and Darfomal (2002, 2003). However, the error across the whole phase space of the discrete equation needs to be understood when any type of adaption is applied to the grid, as it is not always clear which order variables should be refined (or de-refined). Adjoint solutions are useful for deriving errors because they provide information on the first-order sensitivities of a functional (or eigenvalue) to the forward solution of a partial differential equation. The sensitivity information provided depends on how the source term of the adjoint equation is defined; for example, if one seeks to obtain eigenvalue sensitivity to the forward solution, then the eigenvalue must be differentiated with respect to the eigenvector. The adjoint solution will then describe how a small perturbation in the forward solution effects the eigenvalue. Since the computational solution may be regarded as a perturbation from the true solution to the underlying problem, one can use the adjoint equation to derive improvements to the eigenvalue. The approach developed in the present section is an adjoint-based *a posteriori* scheme that derives an approximation to the error in the eigenvalue. This is a similar strategy to that developed for bulk functionals

in Giles et al. (2004) and to that developed for anisotropic grid optimisation in viscous flow Venditti and Darfomal (2000). Focusing on this approach, the current section is arranged as follows. In section (6.2) the eigenvalue that is to be corrected is introduced as a functional. The eigenvalue itself is used later in the section to define an approximation to the error based on forward and locally refined adjoint solutions to the discrete equation. Two normalisations of the eigenvector are considered that simplify the residual. One of these normalisations is based on the fission source and the second is based on the eigenvector. The constrained derivative of the eigenvalue is introduced. In section (6.3), the correction procedure is introduced. This involves expanding the eigenvalue and residual in a first order Taylor series, and an approximation to the error in the eigenvalue is derived. In section (6.4), it is described how one may calculate the error measure in the continuum. In section (6.5) the adjoint problem is introduced. This expresses the eigenvalue in terms of the adjoint system of equations. Due to Galerkin orthogonality, one must used an enriched (improved) estimate of the low-order eigenvector that is placed in the defect equation, otherwise one will find that zero correction is obtained. This is explained in section (6.6), and a scheme for performing this enrichment that works well in diffusive problems is described, that one must apply to the eigenvector before inserting it into the defect equation. In section (6.7) results are demonstrated on a series of optically thick problems, that relate the correction procedure to particle diffusion applications. Conclusions are drawn in section (6.8) with recommendations for further work. These recommendations include application of the correction procedure to hyperbolic problems, and use of the scheme with discontinuous Galerkin finite elements in transport applications.

6.2. Eigenvalue Functional

An approach to eigenvalue correction in particle transport diffusion simulations is considered. The aim is to use the eigenvalue directly in order to derive a defect correction based on the adjoint and a locally refined primal solution. It is also possible to use this correction to automate mesh adaptivity.

6.2.1. Definition of the Eigenvalue

This section describes one possible technique for eigenvalue correction based on the definition of a functional $F(\Psi)$ of the eigenvector Ψ. In the case of an eigen-

value one can write the residual of the governing equation as:

$$\mathscr{R}(\Psi_i) = (\mathbf{A} - \lambda_i \mathbf{B})\Psi_i = \mathbf{0}, \tag{6.1}$$

where λ_i and Ψ_i are the ith generalised eigenvalue and eigenvector respectively, of the matrix $\mathbf{A}$ with respect to the matrix $\mathbf{B}$. $\mathbf{0}$ is a vector of length $\mathcal{N}$ containing zeroes, where $\mathcal{N}$ is the number of unknowns in the problem. This expression can be simplified by normalising the eigenvectors, which is a traditional approach when solving eigensystems. The type of normalisation used for this depends on the physical system. For eigenvector normalisation (EN), one can write:

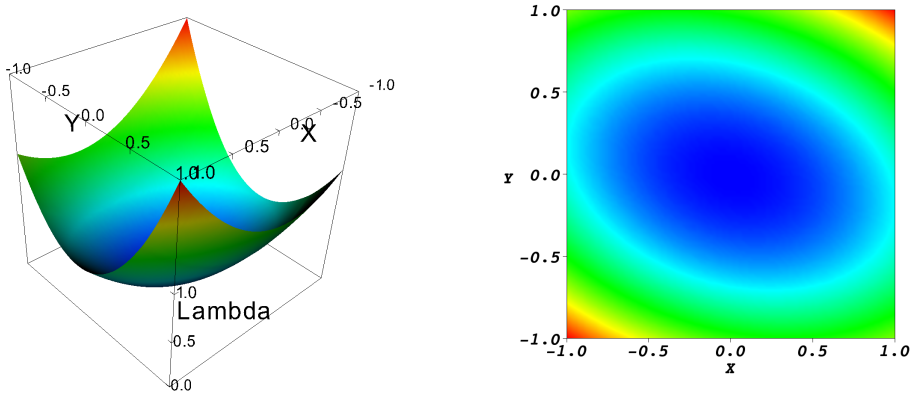
$$C_{\mathrm{EN}}(\Psi_i) = \Psi_i^T \mathbf{B} \Psi_i = 1, \tag{6.2}$$

in which C_{EN} is referred to as the Eigenvector Constraint. Substituting this constraint into equation (6.1) one may write the functional as:

$$F_{\mathrm{EN}}(\Psi_i) = \lambda_i = \Psi_i^T \mathbf{A} \Psi_i. \tag{6.3}$$

However, the associated functional has a saddle point at each of the eigenvalues i.e. the functional has zero first derivatives for a symmetric (self adjoint) operator. One may demonstrate this by choosing two mutually independent vectors Ψ_1 and Ψ_2. A third vector $\Psi_3 = \Psi_3(\Psi_1, \Psi_2)$ is then obtained that satisfies the constraint equation (6.2). Eigenvalues are then generated by substituting Ψ_3 into equation (6.3). The result is illustrated in figure (6.1) to demonstrate the saddle point. Consequently, in this case there is no information on the first order sensitivity of eigenvalues to the components of the eigenvector and one must include quadratic terms in order to apply a defect correction to this functional. Approaches to this are not pursued in the present work. Matrices containing high order derivatives of the eigenvalue would need to be computed in order to implement a quadratic correction, and this is believed to be numerically unfeasible. Use of eigenvector normalisations are therefore not recommended with the present scheme. In order to derive a non-zero linear correction, an alternative form for the normalisation of the eigenvectors is considered, based on the fission source accumulated over the problem. This normalisation is used instead of that defined in equation (6.2), and is defined as:

$$C_{\mathrm{FS}}(\Psi_i) = \mathbf{b}^T \mathbf{B} \Psi_i = 1, \tag{6.4}$$



(a) Surface (b) Contour

Figure 6.1.: Eigenvector normalisation gives rise to a saddle point.

in which C_{FS} is referred to as the Fission Source Constraint, and $\mathbf{b} = [1, 1, ...1]$ is the unit vector of length $\mathcal{N}$ which performs the integration over the problem domain. The functional associated with the fission source normalisation is thus:

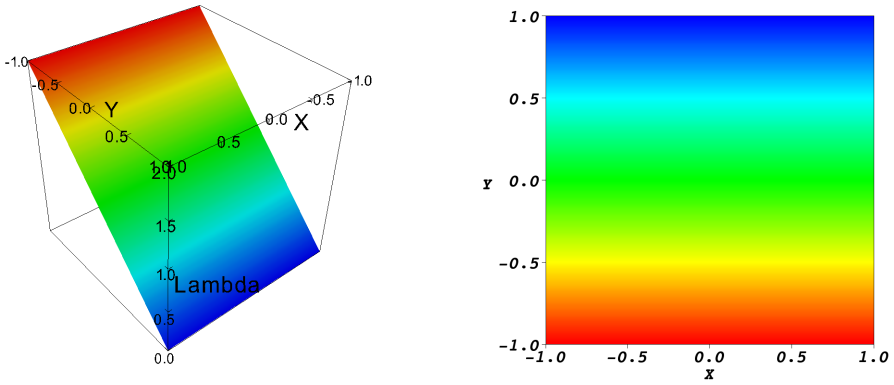
$$F_{\mathrm{FS}}(\Psi_i) = \lambda_i = \mathbf{b}^T \mathbf{A} \Psi_i. \tag{6.5}$$

For clarity the subscript i is dropped in the remainder of the section. The eigenvalue resulting from this normalisation is shown in figure (6.2). In this case, the first-order derivatives of λ are non-zero, and all higher order derivatives are zero, as seen from figure (6.2) which shows there is no curvature when this normalisation is applied. Application of this normalisation therefore results in the first order Taylor expansion being exact because all higher derivatives are zero. Computable linear corrections can thus be found that do not require high order derivatives, such as the Hessian, to be computed.

6.2.2. Derivatives of the Eigenvalue

In order to differentiate the eigenvalue defined in equation (6.5) subject to the corresponding normalisation constraint defined in equation (6.4) we differentiate the ratio of the functional to the constraint:

$$\frac{\partial}{\partial \Psi} \left(\frac{F_{\mathrm{FS}}(\Psi)}{C_{\mathrm{FS}}(\Psi)} \right). \tag{6.6}$$



(a) Surface (b) Contour

Figure 6.2.: Fission source normalisation gives rise to a linear correction.

The constraint is then used to eliminate the terms which appear in the denominator of the resulting expression. Note this procedure can be repeated in order to derive the higher order derivatives of the functional.

6.3. Eigenvalue Correction Procedure

The proposed procedure for recovering the error in the eigenvalue is linear because a first order Taylor series is used. This is exact when the fission-source normalisation is applied, because the high order derivatives of the eigenvalue are all zero. Note this is not the case when the eigenvector normalisation is applied. In that case, high order derivatives exist and would need to be included in order to obtain a correction. This generally requires the Hessian to be computed which is numerically demanding. Worse still, with the eigenvector normalisation there is no information on the first order term because this is zero at a saddle point. Use of the eigenvector normalisation is therefore not recommended.

6.3.1. Linear Functional Correction Procedure

One can expand the functional as a Taylor series as follows:

$$\begin{aligned} F_{\mathrm{FS}}(\Psi_{\mathrm{exact}}) - F_{\mathrm{FS}}(\Psi) &= \left.\frac{\partial F_{\mathrm{FS}}}{\partial \Psi}\right|_{C_{\mathrm{FS}}(\Psi)=1} (\Psi_{\mathrm{exact}} - \Psi) \\ &= \mathbf{b}^T(\mathbf{A} - \lambda\mathbf{B})\,(\Psi_{\mathrm{exact}} - \Psi)\,, \end{aligned} \tag{6.7}$$

in which Ψ_{exact} is the exact eigenvector corresponding to the associated exact eigenvalue that we are attempting to determine. All other terms in this expansion vanish provided that both Ψ and Ψ_{exact} satisfy the normalisation condition equation (6.4), because the condition ensures all derivatives higher than first order are zero in the Taylor series as discussed in section (6.2.1). Likewise, one can expand the residual to first order and obtain:

$$\begin{aligned} \mathscr{R}(\Psi_{\mathrm{exact}}) - \mathscr{R}(\Psi) &= -\mathscr{R}(\Psi) \approx \frac{\partial \mathscr{R}}{\partial \Psi}\,(\Psi_{\mathrm{exact}} - \Psi) \\ &= \left(\mathbf{A} - \lambda\mathbf{B} - (\mathbf{B}\Psi)\mathbf{b}^T(\mathbf{A} - \lambda\mathbf{B})\right)(\Psi_{\mathrm{exact}} - \Psi) \\ &= (\mathbf{A} - \lambda\mathbf{B} - \mathbf{Q})\,(\Psi_{\mathrm{exact}} - \Psi). \end{aligned} \tag{6.8}$$

The matrix $\mathbf{Q}$ has been introduced to simplify the expression, and is defined as $\mathbf{Q} = (\mathbf{B}\Psi)\mathbf{b}^T(\mathbf{A} - \lambda\mathbf{B})$. This contains an outer product, which in contrast to the inner product, generates a full matrix. This acts as an energy sink in the equations serving to make them singular. It is shown later how $\mathbf{Q}$ may be eliminated from the right hand side of the above equation using a special normalisation for the adjoint eigenvector. One can now invert this expression in order to produce the following correction to the eigenvector:

$$\Psi_{\mathrm{exact}} - \Psi = -\left(\frac{\partial \mathscr{R}}{\partial \Psi}\right)^{-1} \mathscr{R}(\Psi). \tag{6.9}$$

The corresponding correction to the eigenvalue is given by:

$$\begin{aligned} F_{\mathrm{FS}}(\Psi_{\mathrm{exact}}) - F_{\mathrm{FS}}(\Psi) &= -\left.\frac{\partial F_{\mathrm{FS}}}{\partial \Psi}\right|_{C_{\mathrm{FS}}(\Psi)=1} \left(\frac{\partial \mathscr{R}}{\partial \Psi}^{-1}\right) \mathscr{R}(\Psi) \\ &= \mathscr{R}(\Psi)^T \left(\frac{\partial \mathscr{R}}{\partial \Psi}\right)^{-T} \left(-\left.\frac{\partial F_{\mathrm{FS}}}{\partial \Psi}\right|_{C_{\mathrm{FS}}(\Psi)=1}\right)^T \\ &= \mathscr{R}(\Psi)^T \Psi^*, \end{aligned} \tag{6.10}$$

where the adjoint flux Ψ^* has been introduced. This is defined as the solution vector to the system:

$$\frac{\partial \mathscr{R}}{\partial \Psi}^T \Psi^* = \left(-\frac{\partial F_{\mathrm{FS}}}{\partial \Psi}\bigg|_{C_{\mathrm{FS}}\Psi=1} \right)^T, \tag{6.11}$$

which may be explicitly written as:

$$\begin{aligned} (\mathbf{A}^T - \lambda \mathbf{B}^T - \mathbf{Q})\Psi^* &= (\mathbf{A}^T - \lambda \mathbf{B}^T)\Psi^* - (\mathbf{A} - \lambda \mathbf{B})^T \mathbf{b} (\mathbf{B}\Psi)^T \Psi^* \\ &= -(\mathbf{A}^T - \lambda \mathbf{B}^T)\mathbf{b}. \end{aligned} \tag{6.12}$$

This serves to define the adjoint eigenvector up to an unknown constant, so one might choose $(\mathbf{B}\Psi)^T\Psi^* = 0$ so that the term containing the $\mathbf{Q}$ matrix can be dropped from the left hand side of the adjoint equation, thereby removing the explicit dependence on Ψ from this equation. When it comes to computing the correction, the Galerkin orthogonality properties of the discretisation ensure that the correction is invariant to a constant offset in the adjoint eigenvector solution, that is to say $\mathscr{R}(\Psi)^T\mathbf{b} = 0$. The resulting adjoint equation is:

$$(\mathbf{A}^T - \lambda \mathbf{B}^T)\Psi^* = -(\mathbf{A}^T - \lambda \mathbf{B}^T)\mathbf{b}. \tag{6.13}$$

Note equation (6.13) may alternatively be written as:

$$(\mathbf{A}^T - \lambda \mathbf{B}^T)(\Psi^* + \mathbf{b}) = \mathbf{0}. \tag{6.14}$$

Consequently, $\Psi^* + \mathbf{b}$ is a generalised eigenvector of the operator $\mathbf{A}^T$ with respect to the matrix $\mathbf{B}^T$ corresponding to the eigenvalue λ, scaled to satisfy $(\mathbf{B}\Psi)^T(\Psi^* + \mathbf{b}) = 1$. For a symmetric (self adjoint) pair of operators this implies that $\Psi^* = \alpha\Psi - \mathbf{b}$ where α is a constant which must be determined from the normalisation condition:

$$0 = (\mathbf{B}\Psi)^T\Psi^* = (\mathbf{B}\Psi)^T(\alpha\Psi - \mathbf{b}) = \alpha\Psi^T\mathbf{B}\Psi - 1, \tag{6.15}$$

which implies that $\Psi^* = (\Psi^T\mathbf{B}\Psi)^{-1}\Psi - \mathbf{b}$. For asymmetric operators one must solve the above equation in order to determine the adjoint eigenvector. Finally, one is in a position to update the functional using the following expression:

$$F_{\mathrm{FS}}(\Psi_{\mathrm{exact}}) = F_{\mathrm{FS}}(\Psi) + \mathscr{R}(\Psi)^T\Psi^*. \tag{6.16}$$

If the same spatial basis functions are used for the forward and adjoint solutions, in conjunction with a Galerkin weighted discretisation of the forward and adjoint problems, then the Galerkin orthogonality properties of the residual will lead to zero correction. Consequently, one must use a more finely resolved and/or higher order solution for the adjoint eigenvectors appearing in the correction term. In the present work this shall be referred to as $\Psi^*_{h/2}$ in which h refers to a mesh spacing. This results in a correction term that is computable:

$$\begin{aligned} F_{\mathrm{FS}}(\Psi_{\mathrm{exact}}) &= F_{\mathrm{FS}}(\Psi) + \mathscr{R}_{h/2}(\Psi_h)^T(\Psi^*_{h/2} - \Psi^*_h) \\ &= F_{\mathrm{FS}}(\Psi) \\ &+ (\Psi^*_{h/2} - \Psi^*_{h\to h/2})^T(\mathbf{A}_{h/2} - \lambda_h \mathbf{B}_{h/2})\Psi_{h\to h/2}, \end{aligned} \tag{6.17}$$

where $h \to h/2$ represents the prolongation of the solution onto the finer mesh and/or higher order basis functions denoted by $h/2$. Typically, the finely resolved adjoint solution would be obtained via use of an C^1 continuous bicubic spline.

6.4. Continuum Error Measure

One can show that equations (6.17) and (6.16) are in fact equivalent, using the definition of the adjoint as introduced by (Riesz, 1907, 1909). In operator form, the adjoint may be defined from the inner product:

$$< \mathcal{L}^*(\psi^*), \psi >=< \psi^*, \mathcal{L}(\psi) > . \tag{6.18}$$

In integral form, this may be written:

$$\int_V dV \mathcal{L}^*(\psi^*)\psi = \int_V dV \psi^* \mathcal{L}(\psi), \tag{6.19}$$

in which $\mathcal{L}^*$ is the adjoint of the operator $\mathcal{L}$ in the continuum. The exact solution in the continuum is represented by ψ_{exact}, and the approximation to this is represented by ψ. These may be regarded respectively as an exact and an approximate solution to the underlying PDE. Thus the exact ith eigenvalue may be written as $\lambda_{exact}(\psi_{exact})$ and an approximate ith eigenvalue as $\lambda(\psi)$.

The error in the approximate eigenvalue with respect to the exact eigenvalue may then written:

$$\lambda_{\mathrm{FS}}(\psi_{exact}) - \lambda_{\mathrm{FS}}(\psi) \quad = \quad \int_V dV \mathscr{R}(\psi)^T (\psi^*_{exact} - \psi^*), \tag{6.20}$$

in which $\mathscr{R}(\psi)$ is the residual in the continuum, ψ^* the adjoint eigenvector in the continuum and ψ^*_{exact} the exact adjoint eigenvector in the continuum. Expanding the right hand side of this equation and using the definition of the adjoint in equation (6.19), one can write:

$$\begin{aligned} \lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) \quad - \quad & \lambda_{\mathrm{FS}}(\psi) \\ = \quad & \int_V dV \mathscr{R}(\psi)^T \psi^*_{exact} - \int_V dV \mathscr{R}(\psi)^T \psi^* \\ = \quad & \int_V dV \psi \mathscr{R}^*(\psi^*_{exact})^T - \int_V dV \psi \mathscr{R}^*(\psi^*)^T, \end{aligned} \tag{6.21}$$

in which $\mathscr{R}^*$ is the adjoint residual. Using the fact that $\mathscr{R}^*(\psi^*_{exact}) = 0$ the above equation becomes:

$$\lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) - \lambda_{\mathrm{FS}}(\psi) \quad = \quad - \int_V dV \psi \mathscr{R}^*(\psi^*)^T. \tag{6.22}$$

This defect may be evaluated numerically by expanding the residual and applying integration by parts to the advection terms. However, this introduces another first derivative and so a second integration by parts needs to be applied for the continuum error to obtained computationally. The procedure for accomplishing this is as follows:

$$\begin{aligned} \lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) \quad - \quad & \lambda_{\mathrm{FS}}(\psi) = - \int_V dV \psi^* \mathscr{R}(\psi)^T \\ = \quad & - \int_V dV \psi^* (\mathbf{A} \cdot \nabla \psi + \mathbf{H}\psi - \lambda \mathbf{B}\psi) \\ = \quad & \int_V dV \left((\nabla \psi^* \cdot \mathbf{A})\psi - \psi^* \mathbf{H} \psi + \psi^* \lambda \mathbf{B} \psi \right) \\ - \quad & \oint_{\Gamma_{IN}} d\Gamma_{IN} \mathbf{n} \cdot \mathbf{A} \psi^*_i \psi - \oint_{\Gamma_{OUT}} d\Gamma_{OUT} \mathbf{n} \cdot \mathbf{A} \psi^* \psi, \end{aligned} \tag{6.23}$$

in which the first integration by parts has been applied to the advection term inside the residual, and the resulting surface integral on the edge Γ of the domain has been split into an inbound and an outbound component. The normal to the domain surface is denoted by the vector $\mathbf{n}$. In two dimensions, the normal to the face of the domain might be written $\mathbf{n} = (n_x, n_y)^T$. The integral on Γ_{IN} is evaluated on the incoming boundary, that is for $\mathbf{n} \cdot \mathbf{A} \leq 0$ and the integral on Γ_{OUT} is evaluated on the outbound surface for $\mathbf{n} \cdot \mathbf{A} > 0$. The adjoint eigenvector on the incoming boundaries is denoted by ψ_i^*. This is to indicate that it contains incoming information from the downwind neighbouring element. The forward and adjoint eigenvectors on the outward boundary are taken from the interior of element, as is the forward eigenvector that appears in the inbound integral. In order to remove the derivative in equation (6.23), that is to remove the derivative of the adjoint eigenvector that appears in the volume integral, one integrates the volume term in the above integral by parts, that is to apply a second integration by parts to the volume term on the rhs of equation (6.23):

$$\begin{aligned}
&\int_V dV \left((\nabla \psi^* \cdot \mathbf{A}) \, \psi - \psi^* \mathbf{H} \psi + \psi^* \lambda \mathbf{B} \psi \right) \\
&= - \int_V dV \left((\mathbf{A} \cdot \nabla \psi) \psi^* - \psi^* \mathbf{H} \psi + \psi^* \lambda \mathbf{B} \psi \right) \\
&+ \oint_{\Gamma_{IN}} d\Gamma_{IN} \mathbf{n} \cdot \mathbf{A} \psi_i^* \psi + \oint_{\Gamma_{OUT}} d\Gamma_{OUT} \mathbf{n} \cdot \mathbf{A} \psi^* \psi,
\end{aligned} \tag{6.24}$$

where in this case it is the forward eigenvector that provides the incoming information on Γ_{IN}. The value of the forward eigenvector that appears in the integrals on the incoming face is assigned the subscript i to indicate that is incoming information that is coming in from the downwind neighbouring element. The value of the adjoint eigenvector ψ^* that appears in the incoming integral is taken from the interior element. Substituting equation (6.24) into equation (6.23) one obtains the

expression:

$$\lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) - \lambda_{\mathrm{FS}}(\psi) = -\int_V dV\,((\mathbf{A}\cdot\nabla\psi)\psi^* - \psi^*\mathbf{H}\psi + \psi^*\lambda\mathbf{B}\psi) + \oint_{\Gamma_{IN}} d\Gamma_{IN}\mathbf{n}\cdot\mathbf{A}\psi^*\psi_i + \oint_{\Gamma_{OUT}} d\Gamma_{OUT}\mathbf{n}\cdot\mathbf{A}\psi^*\psi - \oint_{\Gamma_{IN}} d\Gamma_{IN}\mathbf{n}\cdot\mathbf{A}\psi_i^*\psi - \oint_{\Gamma_{OUT}} d\Gamma_{OUT}\mathbf{n}\cdot\mathbf{A}\psi^*\psi. \quad (6.25)$$

The two outbound surface integrals on Γ_{OUT} in the above equation cancel, simplifying the expression to:

$$\lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) - \lambda_{\mathrm{FS}}(\psi) = -\int_V dV\,((\mathbf{A}\cdot\nabla\psi)\psi^* - \psi^*\mathbf{H}\psi + \psi^*\lambda\mathbf{B}\psi) + \oint_{\Gamma_{IN}} d\Gamma_{IN}\mathbf{n}\cdot\mathbf{A}\psi^*\psi_i - \oint_{\Gamma_{IN}} d\Gamma_{IN}\mathbf{n}\cdot\mathbf{A}\psi_i^*\psi. \quad (6.26)$$

This may be written as:

$$\lambda_{\mathrm{FS}}(\psi_{\mathrm{exact}}) - \lambda_{\mathrm{FS}}(\psi) = -\int_V dV\mathscr{R}(\psi)\psi^* + \oint_{\Gamma_{IN}} d\Gamma_{IN}\mathbf{n}\cdot\mathbf{A}(\psi^* - \psi_i^*)\psi. \quad (6.27)$$

The rhs of equation (6.27) can readily be evaluated numerically, given the forward and adjoint solution vectors. This is done using a numerical integration rule, for example Gaussian quadrature with a sufficient number of points to perform the integral. Higher order than usual may be required, as the function to be integrated contains products of a number of terms which serves to increase the order of the function in the integrand.

6.5. Adjoint Problem

One could also define a functional from the adjoint eigenvalue:

$$F^*_{\mathrm{FS}}(\Psi^*) = \lambda^* = \mathbf{b}^T\mathbf{A}^T\Psi^*\ (=\lambda), \quad (6.28)$$

with the following constraint:

$$C^*_{\mathrm{FS}}(\Psi^*) = \mathbf{b}^T\mathbf{B}^T\Psi^* = 1. \tag{6.29}$$

Repeating the previous derivation with the transpose operators, interchanging the definition of the forward and adjoint solutions, leads to the following procedure. One starts by solving the adjoint problem with the above normalisation of the adjoint eigenvector Ψ^*:

$$\mathscr{R}^*(\Psi^*) = (\mathbf{A}^T - \lambda^*\mathbf{B}^T)\Psi^* = \mathbf{0}. \tag{6.30}$$

One then solves the associated primal problem:

$$(\mathbf{A} - \lambda\mathbf{B})(\Psi + \mathbf{b}) = \mathbf{0}, \tag{6.31}$$

subject to the normalisation $(\mathbf{B}^T\Psi^*)^T\Psi = 0$ and then computes the correction to the adjoint eigenvalue:

$$F^*_{\mathrm{FS}}(\Psi^*_{\mathrm{exact}}) = F^*_{\mathrm{FS}}(\Psi^*) + \mathscr{R}^*(\Psi^*)^T\Psi, \tag{6.32}$$

where an improved forward solution will be required in order to calculate the correction. This procedure will produce a defect correction for the adjoint eigenvalue in agreement with that derived based on the forward solution, provided an identical technique for deriving the higher order solution is used. Note that the alternative pair of adjoint and forward eigenvectors have a different normalisation to that used for the forward correction.

6.6. Local Smoothing

Due to the Galerkin orthogonality, one will find equation (6.16) results in zero correction. An alternative approximation to the error in the eigenvalue is derived in equation (6.17). However, this requires a locally enriched solution to be defined on a fine mesh of spacing $h/2$ and the forward solution, calculated on a coarse mesh of spacing h, to be prolongated on to the fine mesh. The difference between the prolongated solution and the enriched solution may then be used on the fine mesh in order to define a non-zero correction. Typically, a cubic (p=3) polyhedral element type is used to define the enriched solution, and the prolongation operator

$\mathbf{O}(h \longrightarrow h/2)$ projects the linear element solution Ψ_h onto the cubic element to obtain the prolongated solution. This operation may be written as:

$$\Psi_{h \longrightarrow h/2} = \mathbf{O}(h \longrightarrow h/2)\Psi_h, \tag{6.33}$$

in which $\Psi_{h \longrightarrow h/2}$ is the prolongated solution. Note this will be linear in terms of its convergence. Therefore the operator $\mathbf{O}(h \longrightarrow h/2)$ is a 16x4 matrix that operates on the linear solution (which is a vector of length 4) and produces a new vector, $\Psi_{h \longrightarrow h/2}$ which is of length 16 and comprises the prolongated eigenvector values at each of the 16 nodes of the cubic finite element. The improved solution, denoted by $\Psi_{h/2}$, is obtained by evaluating a bicubic surface at the coordinates of these nodes. This is done by taking the linear solution Ψ_h which is known at the 4 corners, and calculating the x and y gradients and the cross-derivative at the 4 corners using finite differences. These quantities are placed in the vector $\mathbf{b}$, of length 16. One then solves the matrix equation:

$$\mathbf{Mc} = \mathbf{b}, \tag{6.34}$$

for the vector of coefficients $\mathbf{c}$. The matrix $\mathbf{M}$ is defined in section (5.5) and is not derived herein. The bicubic surface:

$$S(x, y) = \sum_{j=0}^{3} \sum_{i=0}^{3} c_{ij} x^i y^j, \tag{6.35}$$

is finally evaluated at the 16 nodes to obtain the enriched solution $\Psi_{h/2}$. Fitting a two-dimensional bicubic spline through the eigenvector in this manner produces solutions that are three times differentiable and C^1 continuous across element boundaries. The enriched solution $\Psi_{h/2}$ is therefore much smoother than the underlying solution $\Psi_{h \longrightarrow h/2}$ that bounds it. One may then evaluate the difference between the prolongated and improved solution, that is to obtain $(\Psi_{h/2} - \Psi_{h \longrightarrow h/2})$, at each of the 16 nodes within the cubic element and place this in equation (6.17) to apply defect corrections to the eigenvalue λ.

6.7. Numerical Examples

Results are presented in this section on a series of demanding elliptic problems in which use is made of the fission source normalisation defined in equation (6.4).

All the problems are calculated on a regular orthogonal mesh, as the correction scheme uses a bicubic fit discussed in section (6.6) that is designed for regular spacing of the interpolation points. Correction on unstructured grids is left as topic for future work. The problems are two dimensional in Cartesian geometry, using continuous finite elements. The domain of problem 3 is illustrated in figure (6.1). The first problem comprises two material regions. Region 1 is a pure absorber in

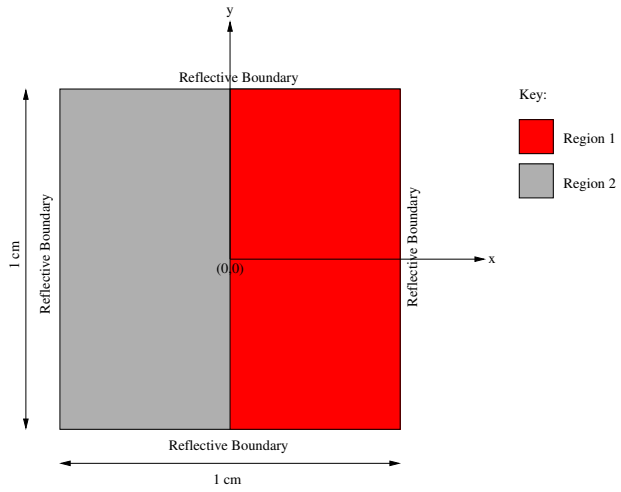


Figure 6.3.: Problem 3 domain.

the range $-0.5 \leq x \leq 0.0, -0.5 \leq y \leq 0.5$. Region 2 contains a fission source of $\sigma_f = 10.0$ and is in the range $0.0 \leq x \leq 0.5, -0.5 \leq y \leq 0.5$. The removal cross-section of both regions is $\sigma_a = 10.0$. The domain of this problem is illustrated in figure (6.3), and the materials used are defined in table (6.1).

Region	σ_a	σ_f
1	10.0	0.0
2	10.0	10.0

Table 6.1.: Definition of materials in problem 3.

Results for problem 3 are shown in figure (6.4), in which eigenvalue error is plotted against number of elements along the x-axis for improving mesh resolution. The

eigenvalue error is defined as $\epsilon = \frac{\lambda - \lambda_{exact}}{\lambda}$, in which λ is the eigenvalue obtained on a coarse mesh using linear (p=1) and quadratic (p=2) continuous finite elements and λ_{exact} is that obtained on a very fine mesh using the same method. The error in the uncorrected eigenvalue is calculated for both the p=1 and p=2 element type. The eigenvalue obtained using the p=1 element type is corrected. The error in this corrected result is also plotted. Correction dramatically improves the convergence rate of the p=1 eigenvalue estimate, by recovering a high proportion of the error on the coarse mesh. The corrected result does not converge as fast as the p=2 eigenvalue, but is significantly better than the p=1 eigenvalue without correction. The error is obtained using the enriched solution, based on a bicubic interpolation as discussed in section (6.6). Correction increases the magnitude of the slope from

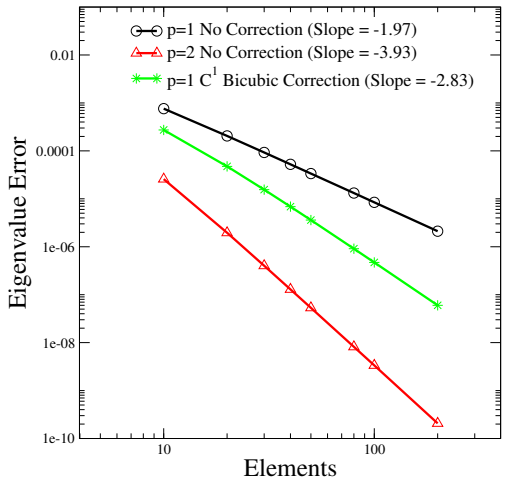


Figure 6.4.: Problem 3 eigenvalue correction.

1.97 (uncorrected) to 3.93 (corrected), recovering an order of convergence in the eigenvalue.

A second problem comprising a source in the top right corner of the domain is considered as the next test case. In this problem, region 1 contains a fission source of $\sigma_f = 10.0$. This is defined in the ranges $0.0 \leq x \leq 0.5$ and $0.0 \leq y \leq 0.5$. Region 2 comprises an absorber of absorption cross-section $\sigma_f = 10.0$. The domain of problem 4 is illustrated in figure (6.5) The materials used in this problem are defined in table (6.7).

The error in both corrected and uncorrected eigenvalue is plotted against number of elements as done for the previous problem. These plots are included in figure

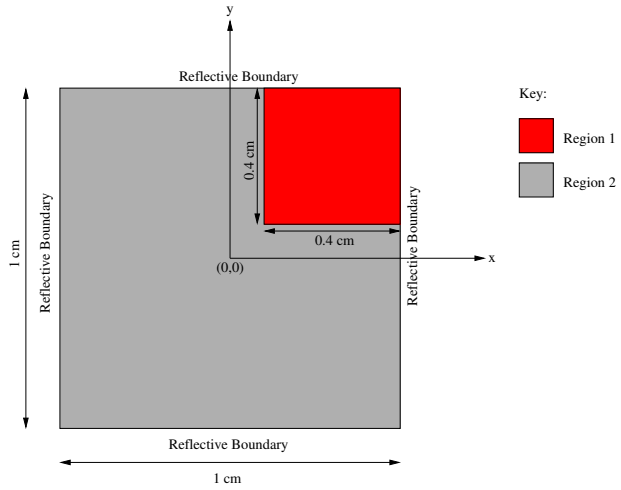
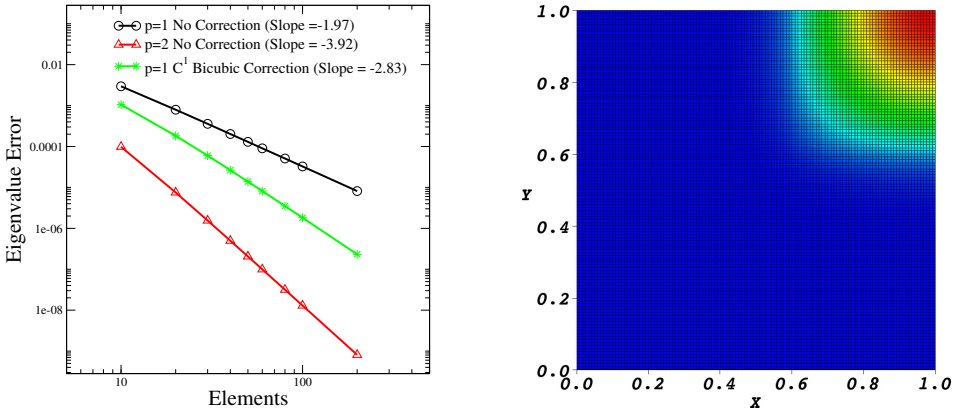


Figure 6.5.: Problem 4 domain.

Region	σ_a	σ_f
1	10.0	10.0
2	10.0	0.0

Table 6.2.: Definition of materials in problem 4.

(6.6a). Uncorrected eigenvalues are shown for both a p=1 and a p=2 element type, and the p=1 corrected eigenvalue is shown. A contour plot of the eigenvalue is included in figure (6.6b) to illustrate the particles being absorbed very sharply as they diffuse from the source region. The gradient in the solution is sharp around the boundary of the source making this a very challenging problem. However, the adjoint based correction scheme performs consistently well in this problem, again recovering an order of convergence with respect to the uncorrected p=1 solution. This improves the magnitude of the slope from 1.97 (p=1 uncorrected) to 3.92 (p=1 corrected). The convergence rate of the corrected eigenvalue is still not as good as that of an uncorrected eigenvalue obtained using a p=2 element. However, a p=2 solution is far more intensive to obtain computationally. A final test problem is considered that is designed to severely challenge the error recovery. Problem 5, the domain of which is illustrated in figure (6.7), comprises four source ”pins”



(a) Correction (b) Eigenvector

Figure 6.6.: Problem 5 correction and eigenvector contour graphic.

inside a moderate absorber.

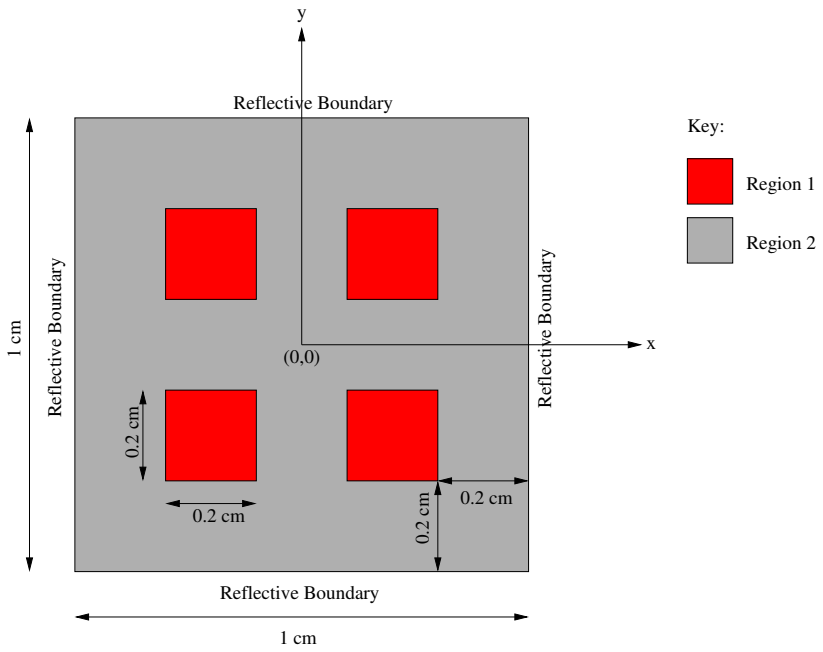


Figure 6.7.: Problem 5 domain.

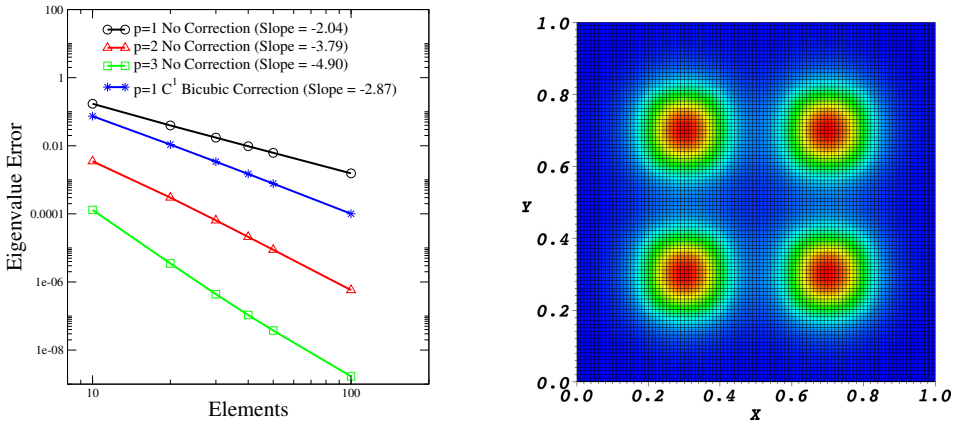
The source pins (region 1) are each of dimension 2 cm x 2cm, and located with their centres positioned at coordinates x,y=(2.5,2.5), (7.5,2.5), (2.5,7.5), (7.5,7.5) within the domain. They material they comprise of has a fission cross-section

$\sigma_f = 10.0$ and absorption cross-section $\sigma_a = 10.0$. The absorber (region 2) has no sources present and an absorption cross-section of $\sigma_a = 10.0$. There is no scattering in this problem. These materials are defined in table (6.7).

Region	σ_a	σ_f
1	10.0	10.0
2	10.0	0.0

Table 6.3.: Definition of materials in problem 5.

Results for problem 5 are presented in figure (6.8). As with the previous problems, eigenvalue error has been plotted against the number of elements along the x-axis of the mesh. The error in the uncorrected eigenvalue is demonstrated for a p=1, p=2 and p=3 element type. The p=1 eigenvalue is corrected with the bicubic spline in the same manner as the previous problems. Correction using the bicubic



(a) Correction (b) Contour

Figure 6.8.: Problem 5 correction and eigenvector contour graphic.

fit again improves the convergence rate of the p=1 eigenvalue, in this test case the magnitude of the slope increases from 2.04 (p=1 uncorrected) to 2.87 (p=1 corrected). This rate is not as good as that of the higher order elements, which in this problem are found to have slope magnitudes of 3.79 (p=2 uncorrected) and

4.90 (p=3 uncorrected). However, use of the p=2 and p=3 element types in this calculation is far more intensive numerically than using a p=1 element with the defect correction, as found with the previous problems. A contour graphic of the eigenvector solution is included to illustrate the profile of the flux in the vicinity of the four pins.

6.8. Conclusions

An adjoint based approach to removing the error in eigenvalue calculations is presented. The scheme, which uses a first order Taylor analysis of the eigenvalue functional and residual of the governing equation, derives an approximation to the error. A defect iteration on the solution is then performed in which the approximation to the error is removed from the solution. Significant improvements to the convergence rate of the eigenvalue are achieved via this approach; the gain is typically one order. This is demonstrated using Galerkin weighted continuous finite elements on a series of demanding two dimensional elliptic problems in Cartesian (X-Y) geometry. The formulation, however, may be applied to hyperbolic problems and other types of discretisation. It may also be applied to other types of functional besides the eigenvalue. Suggested applications of this approach are to define an error metric that may be used to automate mesh adaptivity in addition to improving bulk functional estimates on coarse computational grids.

CHAPTER 7

EIGENVALUE CORRECTION IN HYPERBOLIC PROBLEMS

Synopsis

The eigenvalue correction procedure developed in the previous chapter, in which application to elliptic operators was demonstrated on continuous finite elements, is now applied to transport problems using discontinuous Galerkin weighted finite elements. The transport operator is hyperbolic and capable of shock capture. Therefore the eigenvector solution one obtains is not necessarily smooth; it may only be as smooth as the boundary condition at a material interface. A correction procedure is presented that indicates how to improve the eigenvalue given a better estimate of the eigenvector. It does not, however, indicate how this improved estimate is to be obtained. The present chapter outlines an approach that leads to improved eigenvalues in transport problems using adjoint-based defect estimates.

Contents

7.1. Introduction

Previously, adjoint-based correction procedures were introduced that remove at least part of the error in the eigenvalue. These were found to perform acceptably well when applied to elliptic operators in the context of the diffusion equation, where the underlying computational solution to the eigenvector was continuous. That is, calculated on a mesh expanded using continuous finite elements. The present chapter applies these approaches to hyperbolic operators in the context of the Boltzmann transport equation. In this case the underlying eigenvector solution is obtained using a discontinuous Galerkin finite element discretisation. The need to improve the eigenvalue in transport problems is apparent in many areas of science and mathematics, and has been discussed in chapter (6). Various schemes to this end are the topic of current research. A popular strategy is to control the error contained within the eigenvalue via adaption of the computational grid. Common approaches seek to optimise in some way the number of elements in the spatial discretisation. This strategy improves both the computational efficiency and fidelity of the solution, and usually results in run-time reduction. It has been shown to be a reliable and robust approach in (Baker, 2011) for anisotropic adaption and on meshes composed of triangular elements (Lathowers, 2011). Other popular methods use the adjoint to the forward model to derive an approximation to the error

in the goal or functional, and improve either of these quantities by subtracting this approximation to the error. This approach to error control was applied to eigenvalues in chapter (6) where it was applied to an elliptic operator. In such cases it was found to recover convergence rates in the eigenvalue when a continuous finite element discretisation was applied to the diffusion equation. The underlying scheme for achieving these improvements uses a convolution of the forward and adjoint systems of equations. As discussed previously, due to the Galerkin orthogonality this can lead to zero correction. This requires, therefore, the forward and adjoint systems to be discretised in a different manner ideally one being higher order than the other, to break the orthogonality allowing computable corrections to be derived. In the previous chapter, this is accomplished by enriching either the forward or the adjoint low-order solution, in such a way that incurs the smallest possible overhead. The method presented in chapter (6) was based on a bicubic fit to the underlying solution. This was found to be quite successful in improving the eigenvalue. However, to obtain useful bicubic interpolating splines requires an accurate knowledge of the gradients of the eigenvector. In addition, in two dimensions the cross-derivatives are also needed (in addition to the x and y derivatives) to compute the bicubic coefficients. There is concern over the accuracy of derivative calculations in data that is discontinuous, and previously the method for enriching the eigenvector was only demonstrated for cases where the underlying finite element solution was continuous. The potential this approach has where the underlying finite element solution is discontinuous remains to be proven, and is the topic of the present chapter. It is noted that, even if the gradients of the low-order eigenvector can be accurately obtained within a discontinuous Galerkin framework, the bicubic surfaces would have to be obtained locally. This is because they are not globally obtainable for discontinuous data. The present chapter develops an approach that leads to improvement in the eigenvalue where hyperbolic operators are used and discretised with the discontinuous Galerkin finite element method that has been used throughout this Thesis.

This chapter is arranged as follows. Section (7.2) introduces a method for obtaining the gradient of the discontinuous eigenvector solution. This is useful in deriving an enrichment to the initial, low-order eigenvector so it may be used in the defect recovery procedure. Section (7.3) describes how these gradient are used to smooth the eigenvector solution using a bicubic spline. In section (7.4), a technique for smoothing the eigenvector by fitting the linear element solution to a Serendipity element is described. A third strategy for enriching the low-order

eigenvector is introduced in section (7.5). In section (7.6) the conservation of particles in the system is considered. The schemes described in this chapter generally involve modifying the eigenvector in some way. This will effect the distribution of particles in the system, which in turn has implications for the eigenvalue. In section (7.7), the defect recovery scheme is validated. This is done by placing a solution from a high-order element type into the defect equation, rather than an enriched solution. It is shown that the defect equation recovers the eigenvalue of the high-order solution in this case. The continuum error measure is derived in section (7.8), and shown to be equivalent to the discrete form of the error in section (7.9). In section (7.10) numerical results are shown that demonstrate the various enrichment schemes, and show that one of the schemes leads to eigenvalue improvement. In section (7.11), the chapter concludes with the realisation that although the defect equations indicate how to improve the eigenvalue in transport problems given an enriched eigenvector solution, they do not indicate how this enriched eigenvector is to be obtained. Success of the eigenvalue recovery scheme, therefore depends on how well one can enrich a low order solution.

7.2. Gradient Calculation

This section describes a method for finding the discontinuous gradient of the solution, that is required in order to perform a fit to the low-order eigenvector solution. The enrichment schemes discussed in sections (7.3), (7.4) and (7.5) use this gradient to obtain a smoother result that is intended to approximate a higher-order solution. This is required by the defect equation. The Serendipity fit discussed in section (7.4) is used as an example, in the present section. The approach described here may also be used to obtain the additional cross-derivative terms that are needed to perform the bicubic fit discussed in section (7.3). However, the cross-derivatives are not considered to be very accurate and approaches that do not use the cross-derivative, such as the Serendipity element approach, are preferable to the bicubic approach. This is because the Serendipity system contains only 12 unknowns, and a system of 12 equations for this can be formed without requiring the cross-derivatives. In a three-dimensional problem for example, one may write the gradient as:

$$\nabla\psi(\mathbf{r}) = \left(\frac{\partial\psi(\mathbf{r})}{\partial x}, \frac{\partial\psi(\mathbf{r})}{\partial y}, \frac{\partial\psi(\mathbf{r})}{\partial z}, \right)^T = \mathbf{g}(\mathbf{r}) = \left(g_x(\mathbf{r}), g_y(\mathbf{r}), g_z(\mathbf{r})\right)^T . \quad (7.1)$$

The gradients $\mathbf{g}(\mathbf{r})$ have the usual finite element expansion, that is $g_x(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r}) g_{x_j}$, $g_y(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r}) g_{y_j}$ and $g_z(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r}) g_{z_j}$, in which g_{x_i}, g_{y_i} and g_{z_i} are the x, y, z components of the gradient at node j, respectively. The x gradient may be presented in a weighted residual form, using the finite element basis function $\mathbf{N}_i(\mathbf{r})$:

$$\int_{V_E} dV \mathbf{N}_i(\mathbf{r}) \left(g_x(\mathbf{r}) - \frac{\partial \psi(\mathbf{r})}{\partial x} \right) = 0. \tag{7.2}$$

Inserting the finite element expansion of $g_x(\mathbf{r})$, one obtains:

$$\int_{V_E} dV \mathbf{N}_i(\mathbf{r}) \left(\sum_{j=1}^{\mathcal{N}} \mathbf{N}_i(\mathbf{r}) g_{x_j} - \frac{\partial \psi(\mathbf{r})}{\partial x} \right) = 0, \tag{7.3}$$

in which $g_{x_j}, \forall j \in \{1, 2, \ldots, \mathcal{N}\}$ is the point value of the x component of the gradient, at node j in the finite element mesh. The above equation (7.3) may be expressed as the following matrix equation:

$$\mathbf{M} \underline{\mathbf{g_x}} = \underline{\mathbf{b_x}}, \tag{7.4}$$

in which $\underline{\mathbf{g_x}}$ is a vector of length $\mathcal{N}$ containing the nodal solutions to the gradient in the x direction. That is $\underline{\mathbf{g_x}} = (g_{x_1}, g_{x_2}, \ldots, g_{x_\mathcal{N}})^T$. $\mathbf{M}$ is the $\mathcal{N} \times \mathcal{N}$ mass matrix, the element i, j of which is defined from equation (7.3) to be:

$$\mathbf{M}_{i,j} = \int_{V_E} \mathbf{N}_i(\mathbf{r}) \mathbf{N}_j(\mathbf{r}) dV. \tag{7.5}$$

The vector $\underline{\mathbf{b_x}}$ in the matrix equation (7.4) is of length $\mathcal{N}$ and contains the right hand side of the gradient equations. That is $\underline{\mathbf{b_x}} = (b_{x_1}, b_{x_2}, \ldots, b_{x_\mathcal{N}})^T$. Row i of this vector is defined from the definition of the x component of the gradient to be:

$$\begin{aligned} b_{x_i} = \int_{V_E} dV \mathbf{N}_i(\mathbf{r}) \frac{\partial \psi(\mathbf{r})}{\partial x} &= - \int_{V_E} dV \frac{\partial \mathbf{N}_i(\mathbf{r})}{\partial x} \psi(\mathbf{r}) \\ &+ \oint_{\Gamma_E} d\Gamma n_x \mathbf{N}_i(\mathbf{r}) \psi_{MEAN}(\mathbf{r}), \end{aligned} \tag{7.6}$$

in which integration by parts has been applied to the volume integral, and the resulting surface integral range is the whole of the element surface Γ_E. n_x is

the outward normal to the surface Γ_E in the x direction. The above equation for $\mathbf{b}_{\mathbf{x}i}$ is very straightforward to solve. However, difficulty arises in evaluating the surface integral if the computational solution is discontinuous, because this creates ambiguity concerning the values of $\psi(\mathbf{r})$ on the element boundary Γ_E. One can effectively remove the discontinuities by averaging the values of the solution in the elements on either side of the boundary, and placing the mean average of these values $\psi_{MEAN}(\mathbf{r})$ on the boundary into the integral on Γ_E. That is to take information from the two elements on either side of Γ_E, thereby averaging out the discontinuity. That is to define, for example ψ_{MEAN} in the above equation as:

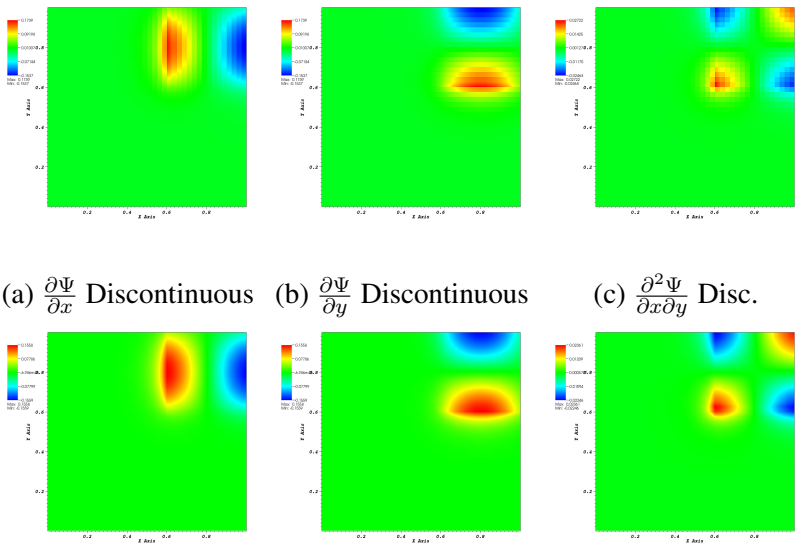
$$\psi_{MEAN}(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i} = \frac{\psi_I(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i} + \psi_O(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_k}}{2}, \tag{7.7}$$

in which $\psi_I(\mathbf{r})$ is the solution to the problem in the element on the inside of surface Γ_E and $\psi_O(\mathbf{r})$ is the solution to the problem in the element on the outside of the surface Γ_E. Node i is a node on the edge of the element inside Γ_E and node k is the neighbouring node on the edge of the element on the outside of Γ_E. Once the vector $\mathbf{b}_{\mathbf{x}}$ has been filled, from the solution of equation (7.6) at each node, the source vector for the matrix equation (7.4) is defined and so equation (7.4) can be solved to obtain the point values of the x gradients, at each node in the element. The mass matrix in this matrix problem makes it a very simple problem to solve; it is small and very efficiently inverted by a direct method. The solution procedure for $\mathbf{g}_{\mathbf{x}}$ may then be repeated for the direction x and y. If the method of averaging the solution on the boundary, that is the definition of $\psi_{MEAN}(\mathbf{r})$ in equation (7.7), is not accurate enough, one may use a more advanced method such as Brezzi-Rabi, which provides a second order approach one might use to calculate gradients of discontinuous fields.

7.3. Bicubic Fit

In section (6), an improvement in the eigenvalue estimate was demonstrated by enriching the eigenvector globally with a C^2 continuous spline. This method of obtaining a best-fit is well proven in image processing and computer graphics, and also has important applications in signal processing. It is applied to the system eigenvector in the present work to derive improvements to the eigenvalue. In two dimensions, this amounts to fitting an interpolating bicubic surface through the existing (known) nodal values. Local smoothness is then added in such a way as to

respect the gradients of the solution. In smooth problems, this accelerates convergence of the eigenvalue (in many cases, quite considerably). A similar approach is developed for DG in the present section. In the case of DG, the discontinuities present considerable difficulty when deriving an interpolating surface. In order to obtain a two dimensional fit to the known data, one must obtain the x and y components of the gradient, and also the cross-derivative. These quantities are required at the nodes of each element on a discontinuous mesh. This is not a well-posed problem; continuity requirements that the gradients have on the underlying solution are not satisfied on these meshes unless some manner of averaging of the nodal values is used. Nodal averaging techniques in this context are discussed in section (7.5). A scheme has been implemented that allows these gradients to be either discontinuous or continuous at the nodes. Both continuous and discontinuous gradients obtained in a sample problem are illustrated for comparison in figure (7.1), which also shows the cross-derivatives. The problem illustrated used a source in the top right corner of the domain inside a heavy absorber, on a 40 $\times$ 40 element mesh. At this resolution, discontinuities are clearly affecting the derivative calculation in



(a) $\frac{\partial \Psi}{\partial x}$ Discontinuous (b) $\frac{\partial \Psi}{\partial y}$ Discontinuous (c) $\frac{\partial^2 \Psi}{\partial x \partial y}$ Disc.

(d) $\frac{\partial \Psi}{\partial x}$ Continuous (e) $\frac{\partial \Psi}{\partial y}$ Continuous (f) $\frac{\partial^2 \Psi}{\partial x \partial y}$ Continuous

Figure 7.1.: Continuous and Discontinuous Gradients of the Eigenvector

(a)-(c). However, the minimum and maximum values of the contours are in very close agreement with the continuous derivative calculation. Once either contin-

uous or discontinuous gradients have been obtained, a bicubic fit is then derived internal to each element. This bicubic fit respects the value of the eigenvector, and the first derivatives of the eigenvector, at the corners of each element. The resulting surface is, therefore itself discontinuous across element boundary. Smoothing of the eigenvector is then achieved locally internal to each element. The bicubic fit provides 16 values internal to the discontinuous element, at the same positions as the nodes in a standard cubic basis function. The fitted element thus approximates a standard cubic finite element but without the cost of solving a global system discretised with a $p = 3$ element type.

7.4. Serendipity Element

The bicubic surface fit described in section (7.3) was shown to be a good method for global enrichment of the eigenvector in cases where the underlying solution is smooth and continuous. There are clearly several disadvantages to this approach when applied to data that contain discontinuities. The main difficulty is the dependence that it has on accurate knowledge of the spatial gradients of the data, in particularly the cross-derivative. These gradients must be obtained to a reasonable accuracy in order to determine reliable coefficients of the interpolating two dimensional spline. Each time one differentiates the solution, the order of the equations are reduced. This is seen as a reduction in the convergence. In addition, the gradients have certain continuity requirements on the underlying eigenvector which are rarely met on element boundaries by discontinuous finite element methods. There is concern, therefore, over nodal representations of the eigenvector gradients

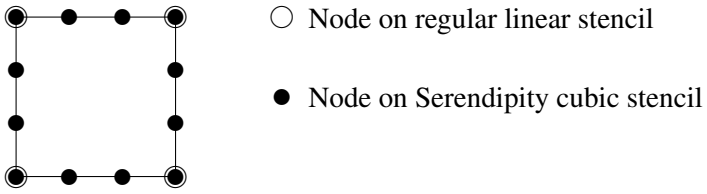
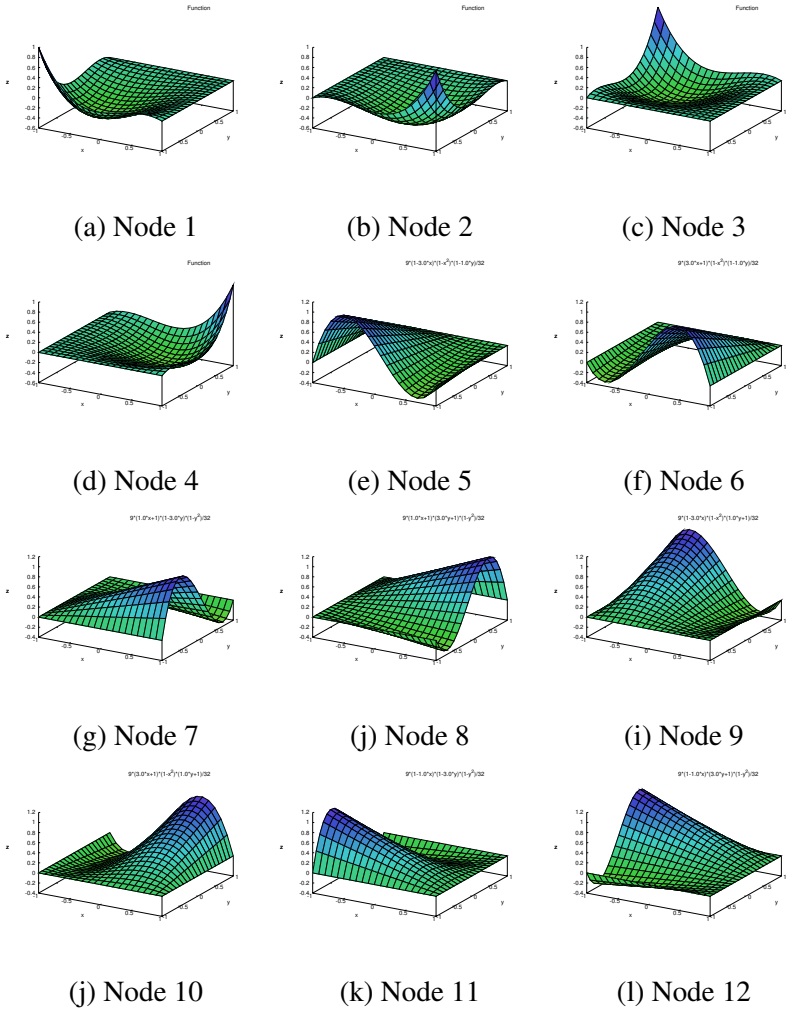


Figure 7.2.: **Serendipity $p = 3$ Element Type**

with DG. Use of a Serendipity element type eliminates the cross-component of the derivative from the enrichment scheme. However, values of both the x and y component are still needed for two dimensional interpolation. With the dependence on the cross-component removed, some improvement to the enrichment process is to

be expected. A cubic Serendipity element is illustrated in figure (7.2). The element has no nodes in the centre, but a cubic variation along the faces. It is popular as a transition element on meshes where the degree of the basis function is adaptive (p refinement) and boundary values flow out of a lower-order element into a higher order element. The shape functions that construct the cubic serendipity element are illustrated in figure (7.3). In the present work, the 4 corner nodes in the cubic



(a) Node 1 (b) Node 2 (c) Node 3

(d) Node 4 (e) Node 5 (f) Node 6

(g) Node 7 (j) Node 8 (i) Node 9

(j) Node 10 (k) Node 11 (l) Node 12

Figure 7.3.: Bases of the Cubic Serendipity Element Type

Serendipity element are numbered in exactly the same manner as the 4 nodes in the standard linear element. The values in each corner (nodes $i = 1, 4$) of the cubic Serendipity element, and their first derivatives in x and y, are therefore the same as those of the standard linear element. They can be used to form a system of

equations in which there are 8 unknowns which are the remaining unknown nodal values of the Serendipity element. One does not need the cross-derivative to solve this system. A matrix equation for these unknown nodal values in the Serendipity element is now derived. The solution at position $\mathbf{r}$ in the cubic Serendipity element has the finite element representation:

$$\psi^s(\mathbf{r}) = \sum_{j=1}^{12} \mathbf{S}_j(\mathbf{r})\phi_j^s, \tag{7.8}$$

in which $\mathbf{S}_i(\mathbf{r}), \forall i \in \{1, 2, \ldots, 12\}$ is the node i cubic Serendipity finite element shape function at position $\mathbf{r}$, and $\phi_i^s, \forall i \in \{1, 2, \ldots, 12\}$ is the computational finite element solution at node i. The 12 nodes in the cubic Serendipity element are distributed on the surface of the element, 4 of which (the corner nodes) are at the same location as the 4 nodes in the standard linear element. Therefore, one may equate the x derivative of the Serendipity element solution $\psi^s(\mathbf{r})$ at the corner nodes $i, \forall i \in \{1, 2, 3, 4\}$ in the Serendipity element, to the x derivative of the standard linear element solution at the 4 nodes in the standard linear element. The method for obtaining these values is described in section (7.2). That is to write the x derivative of the cubic Serendipity element solution as:

$$\left.\frac{\partial \psi^s(\mathbf{r})}{\partial x}\right|_{\mathbf{r}=\mathbf{r}_i} = g_x(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i}, \tag{7.9}$$

in which the position of corner node i is denoted by $\mathbf{r}_i, \forall i \in \{1, 2, 3, 4\}$ and the function $g_x(\mathbf{r})$ describes the x component of the linear element solution gradient at position $\mathbf{r}$. This may be calculated at the nodes $i, \forall i \in \{1, 2, 3, 4\}$ from the linear element solution using the method described in section (7.2). The y component of the cubic Serendipity element solution is, likewise, equated to the y component of the derivative of the standard linear element solution at the corners:

$$\left.\frac{\partial \psi^s(\mathbf{r})}{\partial y}\right|_{\mathbf{r}=\mathbf{r}_i} = g_y(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i}, \tag{7.10}$$

in which the function $g_y(\mathbf{r})$ describes the y component of the linear element solution gradient at position $\mathbf{r}$ and $\forall i \in \{1, 2, 3, 4\}$. One may repeat the above for higher dimensions, such as z and time in a more general problem. Note that at the 4 corner nodes of the Serendipity element, the values of $\psi^s(\mathbf{r})$ may be equated to

those of the linear element solution:

$$\psi^s(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i} = \psi(\mathbf{r})|_{\mathbf{r}=\mathbf{r}_i}\,, \tag{7.11}$$

in which $\psi(\mathbf{r})$ is the solution to the standard linear element and $\mathbf{r}_i, \forall i \in \{1,2,3,4\}$ is the position of node i, that is common to both the standard linear element and the cubic Serendipity element. Once the x and y derivatives have been obtained from the linear element solution using the method described in section (7.2), they may be placed in the vectors $\mathbf{g_x}$ and $\mathbf{g_y}$. One may then write:

$$\sum_{j=1}^{12} \frac{\partial \mathbf{S}_j(\mathbf{r})}{\partial x}\bigg|_{\mathbf{r}=\mathbf{r}_i} \phi_j^s = g_{x_i}, \tag{7.12}$$

where equation (7.8) has been inserted into equation (7.9) and row $i, \forall i \in \{1,2,3,4\}$ of the vector $\mathbf{g_x}$ denoted by g_{x_i}. Repeating this for the y component of the gradient, one obtains:

$$\sum_{j=1}^{12} \frac{\partial \mathbf{S}_j(\mathbf{r})}{\partial y}\bigg|_{\mathbf{r}=\mathbf{r}_i} \phi_j^s = g_{y_i}, \tag{7.13}$$

where g_{y_i} refers to row $i, \forall i \in \{1,2,3,4\}$ of the vector $\mathbf{g_x}$. Repeating the above process for the values of the solution on the corner nodes $i, \forall i \in \{1,2,3,4\}$:

$$\sum_{j=1}^{12} \mathbf{S}_j(\mathbf{r})\bigg|_{\mathbf{r}=\mathbf{r}_i} \phi_j^s = \phi_i, \tag{7.14}$$

in which ϕ_i is the standard linear element solution at node i on the standard linear element. Equations (7.12), (7.13) and (7.14) above may be combined into a 12×12 matrix expression that contains all 12 nodal solutions to the cubic Serendipity

element:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 1 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 1 & 0 & ... & 0 \\ a_{5,1} & ... & ... & ... & ... & ... & a_{5,12} \\ a_{6,1} & ... & ... & ... & ... & ... & a_{6,12} \\ a_{7,1} & ... & ... & ... & ... & ... & a_{7,12} \\ a_{8,1} & ... & ... & ... & ... & ... & a_{8,12} \\ a_{9,1} & ... & ... & ... & ... & ... & a_{9,12} \\ a_{10,1} & ... & ... & ... & ... & ... & a_{10,12} \\ a_{11,1} & ... & ... & ... & ... & ... & a_{11,12} \\ a_{12,1} & ... & ... & ... & ... & ... & a_{12,12} \end{pmatrix} \begin{pmatrix} \phi_1^s \\ \phi_2^s \\ \phi_3^s \\ \phi_4^s \\ \phi_5^s \\ \phi_6^s \\ \phi_7^s \\ \phi_8^s \\ \phi_9^s \\ \phi_{10}^s \\ \phi_{11}^s \\ \phi_{12}^s \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ g_{x_1} \\ g_{x_2} \\ g_{x_3} \\ g_{x_4} \\ g_{y_1} \\ g_{y_2} \\ g_{y_3} \\ g_{y_4} \end{pmatrix}, \tag{7.15}$$

in which the vector $\phi^{\mathbf{s}}$ contains the solutions to the cubic Serendipity element, that is $\phi^{\mathbf{s}} = (\phi_1^s, \phi_2^s, \ldots, \phi_{12}^s)^T$ and the vector that forms the right hand side contains the linear element solution and its x and y derivatives arranged into a vector of length 12. In the above matrix equation, element $a_{i,j}, \forall j \in \{1, 2, \ldots, 12\}$ is defined as:

$$a_{ij} = \begin{cases} \mathbf{I}_{ij} & 1 \leq i \leq 4; \\ \left.\frac{\partial \mathbf{S}_j(\mathbf{r})}{\partial x}\right|_{\mathbf{r}=\mathbf{r}_{i-4}} & 5 \leq i \leq 8; \\ \left.\frac{\partial \mathbf{S}_j(\mathbf{r})}{\partial y}\right|_{\mathbf{r}=\mathbf{r}_{i-8}} & 9 \leq i \leq 12, \end{cases} \tag{7.16}$$

in which $\mathbf{I}_{ij}$ is element i, j of the 12×12 identity matrix. The above system is easily solved in-line very efficiently for the 12 nodal values $\phi_i^s, \forall i \in \{1, 2, \ldots, 12\}$ of the Serendipity element. These values might then be used as boundary values on a higher order element, for example a $p = 3$ cubic standard element. Applying this approach to each element on an $\mathcal{O}(h)$ mesh may result in improved eigenvector solution that approximates the solution of an $\mathcal{O}(\frac{h}{2})$. This improved, or enriched estimate can be used in place of the eigenvector obtained from a full-cubic discretisation, at significantly reduced computational cost. The enriched solution is far less intensive than the full-cubic solution, as the only iteration overhead is that of the underlying solution which is linear (on a $p = 1$ element type).

7.5. Sub-Grid Enrichment

Eigenvalue improvement strategies have been found to be highly successful when used on the sub-grid scale (Baker, 2011). These approaches have been developed for anisotropic mesh adaption where the finite element solution has been decomposed into a continuous and discontinuous part (Candy, 2010). The discontinuous part is complementary to the continuous solution. Within a DG framework, it has been suggested that the problems presented by the discontinuities inherent in the scheme can be somewhat avoided if a sub-grid-like strategy is formulated within the discretisation. One way of achieving this is to average the discontinuous values that surround a node (or mesh vertex) into a single value at that point. One then enriches the resulting continuous solution. In practise, it might be more satisfactory to put the discontinuities back in to the continuous solution, after it has been enriched onto a cubic mesh. That is to prolongate the differences between the two solutions on to the same mesh as the enriched solution so they may be subtracted from it, restoring the discontinuities. If done in a consistent manner this retains the enrichment performed on the continuous part. One may write in the continuum, the superposition of the two solutions:

$$\Psi(\mathbf{r}) = \Theta(\mathbf{r}) + I(\mathbf{r}), \tag{7.17}$$

in which $\Psi(\mathbf{r})$ is the solution to be made discrete with discontinuous finite elements and $\Theta(\mathbf{r})$ is a continuous discrete solution. $I(\mathbf{r})$ is simply the difference between the continuous and discontinuous components, and allows the discontinuities to be restored after any post-processing operation on $\Theta(\mathbf{r})$, such as eigenvector enrichment. This decomposition is potentially useful where any such post-processing is more easily performed on continuous data. For example, one might improve the continuous part with the bicubic spline approach used in section (6). This procedure may be written:

$$\hat{\Theta(\mathbf{r})} = \mathcal{B}\Theta(\mathbf{r}), \tag{7.18}$$

in which $\mathcal{B}$ is the operation that fits a C^2 continuous bicubic spline to the original data, generating a new vector $\hat{\Theta(\mathbf{r})}$ that is more accurate due to this operation. One then obtains an improved discontinuous solution by performing the operation:

$$\hat{\Psi(\mathbf{r})} = \hat{\Theta(\mathbf{r})} + I(\mathbf{r}), \tag{7.19}$$

Typically, the quantity $\hat{\Theta(\mathbf{r})}$ will exist on a finer computation grid than $I(\mathbf{r})$. Therefore $I(\mathbf{r})$ would need to be prolongated linearly onto the same mesh as $\hat{\Theta(\mathbf{r})}$.

7.6. Particle Conservation

In order to obtain an improvement to the eigenvalue, the distribution of particles in the system must be modified. It is understood the system eigenvalue is sensitive to location of each particle in the system, and modification to the system eigenvectors other than their linear combination or scaling will modify the position of particles locally across the system. The computed uncorrected distribution will converge to a profile that is representative of the system. This is because improvements made to the mesh better resolves the number of particles at every point in the domain. Any enrichment operation performed on the eigenvector will not conserve particles locally, but will have no effect on the total number in the system, i.e. the global number of particles in the system will be unaltered by the enrichment procedure even though local conservation is violated.

7.7. Exact Eigenvalue

It is possible to show that the defect expression, equation (6.20), can produce an exact eigenvalue when the exact adjoint eigenvector is placed in this equation. Not only is this an important validation exercise for the defect algorithm within the code, which is based on equation (6.20) it gives one confidence in the theory behind this method for recovering eigenvalue errors. In practice, one would obtain a low-order eigenvector (i.e. from a calculation using linear finite elements) and enrich it using one of the procedures described in the sections above. The enriched value would then be placed into the defect equation to improve the estimate of the eigenvalue. But as a validation exercise for the code, one could instead insert the eigenvector obtained from a calculation using high-order elements in to the defect equation. The defect equation should then recover an eigenvalue identical to that from the high-order element calculation. To demonstrate this, one first expresses the residual in the continuum:

$$\begin{aligned}\mathscr{R}(\psi(\mathbf{r})) &= ((\mathbf{A}\cdot\nabla + \mathbf{H}(\mathbf{r}) - \lambda\mathbf{B}(\mathbf{r}))\psi(\mathbf{r}))^T \\ &= \psi(\mathbf{r})^T(\mathbf{A}\cdot\nabla^T + \mathbf{H}(\mathbf{r})^T - \lambda^*\mathbf{B}(\mathbf{r})^T),\end{aligned} \tag{7.20}$$

in which $\lambda = \lambda(\psi)$ is the ith eigenvalue associated with the ith eigenvector $\psi(\mathbf{r})$, where $\psi(\mathbf{r})$ is the eigenvector at position $\mathbf{r}$ in the continuum, and $\mathbf{A}$, $\mathbf{H}$ and $\mathbf{B}(\mathbf{r})$ are matrices containing the angular discretisation. The adjoint eigenvector at position $\mathbf{r}$ may be written represented as $\psi^*(\mathbf{r})$, which is an approximate solution to the adjoint system of equations. Representing the exact solution to the adjoint system with $\psi_e^*(\mathbf{r})$, one may write the defect in the adjoint eigenvector as $\psi^*(\mathbf{r}) - \psi_e^*(\mathbf{r})$. From this point, the dependence on $\mathbf{r}$ is assumed so that it may be omitted from the notation. Multiplying the continuum residual in the above equation by this defect and integrating over the domain V, the following equation is obtained:

$$\int_V dV \mathscr{R}(\psi)^T(\psi^* - \psi_e^*) = \int_V dV \psi^T((\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^*\mathbf{B}^T)\psi^* - (\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^*\mathbf{B}^T)\psi_e^*). \quad (7.21)$$

From the definition of the adjoint equations, one will note that:

$$(\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^*(\psi^*)\mathbf{B}^T)\psi^* = \mathbf{0}, \quad (7.22)$$

but:

$$(\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^*(\psi^*)\mathbf{B}^T)\psi_e^* \neq \mathbf{0}, \quad (7.23)$$

because $\lambda^* = \lambda^*(\psi^*) \neq \lambda^*(\psi_e^*)$. Equations (7.22) and (7.23) both contain the same residual but in each case operating on a different eigenvector. Using equations (7.21) and (7.22) one readily obtains:

$$\int_V dV \mathscr{R}^T(\psi)(\psi^* - \psi_e^*) = -\int_V dV \psi^T(\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^T\mathbf{B}^T)\psi_e^* = -\int_V dV \psi^T(\mathbf{A}\cdot\nabla^T + \mathbf{H}^T - \lambda^T\mathbf{B}^T)(\psi_e^* + 1), \quad (7.24)$$

since if ψ^* is a solution to the adjoint system of equations then so must be $\psi^* + 1$ be also. This is because the adjoint system is solved for any linear combination of the adjoint solution. Inserting the expression:

$$\left(\mathbf{A}\cdot\nabla^T + \mathbf{H}^T\right)(\psi_e^* + 1) = \lambda^*(\psi_e^*)\mathbf{B}^T(\psi_e^* + 1), \quad (7.25)$$

into equation (7.24) and using the fact that $\lambda = \lambda^*$, it is easily shown:

$$\begin{aligned}
&\int_V dV \mathscr{R}^T(\psi)(\psi^* - \psi_e^*) \\
&= -\int_V dV \psi^T(\lambda^*(\psi_e^*)\mathbf{B}^T(\psi_e^* + 1) - \lambda^*(\psi^*)\mathbf{B}^T(\psi_e^* + 1)) \\
&= -\int_V dV \lambda^*(\psi_e^*)(\psi^T\mathbf{B}^T(\psi_e^* + 1)) + \lambda^*(\psi^*)(\psi^T\mathbf{B}^T(\psi_e^* + 1)). \qquad (7.26)
\end{aligned}$$

From the normalisation of the adjoint, that is the condition $\psi^T\mathbf{B}^T(\psi_e^* + 1) = 1$, the above expression may be written:

$$\int_V dV \mathscr{R}(\psi)(\psi^* - \psi_e^*) = -\lambda^*(\psi_e^*) + \lambda^*(\psi^*), \qquad (7.27)$$

which is an identical expression to equation (6.20) derived from the Taylor expansions of the residual and the eigenvalue. Thus if one places the exact eigenvector into equation (6.20), the exact eigenvalue will be obtained. This has been verified within the code; the defect iteration using equation (6.20) predicts an identical eigenvalue to that of any high-order power iteration when the eigenvector from that power iteration is supplied to the defect iteration.

7.8. Continuum Error Measure

In order to evaluate the approximation to the eigenvalue defect derived in equation (7.27), one must perform a numerical integration in the continuum. This section describes how this should be done for the discontinuous Galerkin method. One first obtains the forward eigenvector $\psi(\mathbf{r}) = \sum_j^{\mathcal{N}} \mathbf{N}_j(\mathbf{r})\phi_j$ on an $\mathcal{O}(h)$ mesh via inverse power iteration, in which $\mathbf{N}_i(\mathbf{r}), i \in \{1, 2, \ldots, \mathcal{N}\}$ is the node i $\mathcal{O}(h)$ finite element shape function at position $\mathbf{r}$ in the problem domain and ϕ_i the computational solution at node i in the $\mathcal{O}(h)$ mesh. This yields the ith (for example the dominant) eigenvalue $\lambda = \lambda(\psi)$. The adjoint equations are then solved in-line on the $\mathcal{O}(h)$ mesh to obtain the adjoint eigenvector $\psi^* = \sum_j^{\mathcal{N}} \phi_j^*$ and adjoint ith eigenvalue $\lambda^* = \lambda^*(\psi^*) = \lambda(\psi)$. The adjoint eigenvector is then enriched using one of the procedures describe above, to produce a higher-order solution $\psi_{\frac{h}{2}}^*$. This enrichment process may be written as an operator acting on the $\mathcal{O}(h)$ solution and

expressed as $\psi^*_{\frac{h}{2}} = \mathcal{E}_{\frac{h}{2}}(\psi^*)$ in which $\mathcal{E}_{\frac{h}{2}}$ is the enrichment operator that takes as its argument the $\mathcal{O}(h)$ solution and generates the higher order result that is $\mathcal{O}(\frac{h}{2})$. This operator performs the action of one of the schemes in the above sections. The equation for the defect, that is equation (7.27), indicates how the eigenvalue λ can be improved given a better estimate $\psi^*_{\frac{h}{2}} \approx \psi^*_e$ of the adjoint eigenvector ψ^*. It does not indicate, however, how this better estimate is to be obtained. Assuming one has obtained $\psi^*_{\frac{h}{2}}$, such that $\psi^*_{\frac{h}{2}}$ is a better approximation to the exact solution of the adjoint equations, the error in the eigenvalue can be approximated by evaluating the integral:

$$\begin{aligned}
\lambda^*_{\frac{h}{2}}(\psi^*_{\frac{h}{2}}) - \lambda^*(\psi^*) = \lambda_{\frac{h}{2}}(\psi_{\frac{h}{2}}) - \lambda(\psi) &= \int_V dV \mathscr{R}(\psi)^T \left(\psi^* - \psi^*_{\frac{h}{2}}\right) \\
&= \int_V dV \left(\mathbf{A}\cdot\nabla\psi + \mathbf{H}\psi - \lambda\mathbf{B}\psi\right)^T \left(\psi^* - \psi^*_{\frac{h}{2}}\right) \\
= \int_V dV \left(-\nabla\left(\psi^* - \psi^*_{\frac{h}{2}}\right)\cdot\mathbf{A}\psi + \mathbf{H}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right) - \lambda\mathbf{B}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right)\right) \\
+ \oint_{\Gamma_{IN}} d\Gamma \left(\psi^* - \psi^*_{\frac{h}{2}}\right)\mathbf{A}\cdot\mathbf{n}\psi_{IN} + \oint_{\Gamma_{OUT}} d\Gamma \left(\psi^* - \psi^*_{\frac{h}{2}}\right)\mathbf{A}\cdot\mathbf{n}\psi. \quad (7.28)
\end{aligned}$$

in which $\lambda_{\frac{h}{2}}(\psi_{\frac{h}{2}})$ is that exact eigenvalue one would obtain given the exact eigenvector $\psi_{\frac{h}{2}}$. Thus $\lambda_{\frac{h}{2}}(\psi_{\frac{h}{2}}) - \lambda(\psi)$ is the error between the exact eigenvalue and the eigenvalue that is available from the low-order solution. In the above expression, integration by parts has been applied to the advection terms in the residual. The surface terms that arise from this have been split into an inbound part and an outbound part. The inbound part of this contains incoming information in the vector ψ_{IN}. This is the forward eigenvector in the neighbouring element. Applying integration by parts a second time to remove the derivative that remains in the volume integral, one obtains:

$$\begin{aligned}
&\int_V dV \left(-\nabla\left(\psi^* - \psi^*_{\frac{h}{2}}\right)\cdot\mathbf{A}\psi + \mathbf{H}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right) - \lambda\mathbf{B}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right)\right) \\
&= \int_V dV \left(\mathbf{A}\cdot\nabla\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right) + \mathbf{H}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right) - \lambda\mathbf{B}\psi\left(\psi^* - \psi^*_{\frac{h}{2}}\right)\right) \\
&\qquad - \oint_{\Gamma_{IN}} d\Gamma \left(\psi^* - \psi^*_{\frac{h}{2}}\right)\mathbf{A}\cdot\mathbf{n}\psi - \oint_{\Gamma_{OUT}} d\Gamma \left(\psi^* - \psi^*_{\frac{h}{2}}\right)\mathbf{A}\cdot\mathbf{n}\psi. \quad (7.29)
\end{aligned}$$

Note that neither surface term in equation (7.29) contains incoming information; both ψ and $\psi^*_{\frac{h}{2}}$ in the surface integrals in the above equation are taken from inside the element across which the integration is performed. They are, however, across different surfaces. Substituting equation (7.29) into equation (7.28) the two integrals over the surface Γ_{OUT} cancel to leave only the inbound surface integrals and the volume term. This reduces the expression to:

$$\lambda_{\frac{h}{2}}(\psi_{\frac{h}{2}}) - \lambda(\psi) = \int_V dV \mathscr{R}(\psi)^T \left(\psi^* - \psi^*_{\frac{h}{2}}\right) - \oint_{\Gamma_{IN}} d\Gamma \left(\psi^* - \psi^*_{\frac{h}{2}}\right) \mathbf{n} \cdot \mathbf{A} \left(\psi - \psi_{IN}\right). \tag{7.30}$$

One then computes this expression using a numerical quadrature rule that has sufficient number of points to respect the order of the integrand. Note that the integrand is at most $\mathcal{O}(6)$ as both ψ and $\psi^*_{\frac{h}{2}}$ are no more than $\mathcal{O}(3)$. An n-point quadrature is required to integrate up to a $2n-1$ degree polynomial. Therefore, a 4-point quadrature rule should be sufficient to perform this integral. Note this will require the integrand in the above to be evaluated at 16 points, for two dimensional elements.

7.9. Discrete Error Measure

It is possible to use the discrete residual, rather than the residual expressed in the continuum, to obtain the defect approximation. As one might expect, these two definitions of the defect are complementary to one another, and can be shown to be equivalent. Therefore, they will yield identical values for the defect estimate, when computed. This section presents the discrete error measure for eigenvalue problems and shows that it is equivalent to the continuum error measure. First, the discrete residual $\mathbf{R}(\mathbf{\Phi})$ that is a function of the vector containing the discrete solution $\mathbf{\Phi}$, is considered. $\mathbf{R}(\mathbf{\Phi})$ is a vector of length $\mathcal{M} \times \mathcal{N}$ where $\mathcal{M}$ is the number of moments in the angular discretisation and $\mathcal{N}$ the number of nodes in the finite element mesh. $\mathbf{\Phi}$ is a vector of length $\mathcal{M} \times \mathcal{N}$ also, containing the computational solutions $\phi_i, i \in \{1, 2, \ldots, \mathcal{M} \times \mathcal{N}\}$ to the system of discrete equations that are implemented in matrix form in the code. Due to the Galerkin orthogonality, typically the discrete residual would be calculated on a mesh of order $\mathcal{O}(\frac{h}{2})$ while the solution to the discrete equations would be calculated on an order $\mathcal{O}(h)$ mesh.

One would therefore prolongate the $\mathcal{O}(h)$ solution onto the $\mathcal{O}(\frac{h}{2})$ mesh in order to insert it into the discrete residual. If one were to place the $\mathcal{O}(\frac{h}{2})$ solution itself into this residual, they would expect it to equate to a vector of length $\mathcal{M} \times \mathcal{N}$ containing zeroes. The discrete residual calculated on an $\mathcal{O}(\frac{h}{2})$ of the $\mathcal{O}(h)$ solution is therefore written as $\mathbf{R}_{\frac{h}{2}}(\mathbf{\Phi}_h)$ in which it is understood that $\mathbf{\Phi}_h$ is the $\mathcal{O}(h)$ prolongated onto the $\mathcal{O}(\frac{h}{2})$ mesh on which the discrete residual is calculated. The defect in the eigenvalue may be written in terms of the discrete residual:

$$\lambda_{\frac{h}{2}} - \lambda_h = \mathbf{R}_{\frac{h}{2}}(\mathbf{\Phi}_h)^T \left(\mathbf{\Phi}^*_h - \mathbf{\Phi}^*_{\frac{h}{2}} \right), \tag{7.31}$$

in which $\mathbf{\Phi}^*_h$ is the vector of length $\mathcal{M} \times \mathcal{N}$ containing the discrete solution to the adjoint system of equations on the $\mathcal{O}_h$ mesh and prolongated onto the $\mathcal{O}_{\frac{h}{2}}$. $\mathbf{\Phi}^*_{\frac{h}{2}}$ is a vector, also of length $\mathcal{M} \times \mathcal{N}$, containing a discrete adjoint solution obtained on the $\mathcal{O}(\frac{h}{2})$ mesh. That is, $\mathbf{\Phi}^*_{\frac{h}{2}}$ is an higher-order solution than $\mathbf{\Phi}^*_h$. The dot product of the two mesh-wise vectors in the above expression produces an estimate of the difference between the two eigenvalues. To show this is in fact equivalent to that obtained by solving equation (7.30), the finite element approximations to $\psi^*_{\frac{h}{2}}(\mathbf{r})$ and $\psi^*(\mathbf{r})$ are inserted into the defect equation (equation (7.30)). That is to place $\psi^*_{\frac{h}{2}}(\mathbf{r}) = \sum_i = \mathbf{N}_i(\mathbf{r})\phi^*_{\frac{h}{2}_i}$ and $\psi^* = \sum_i \mathbf{N}_i \phi^*_{h_i}$ into equation (7.30), in which $\mathbf{N}_i$ is the node i order $\mathcal{O}(\frac{h}{2})$ finite element shape function. One then obtains:

$$\begin{aligned} \lambda_{\frac{h}{2}} - \lambda_h = & \sum_{i=1}^{\mathcal{N}} \int_V dV \mathbf{N}_i \mathscr{R}(\psi_h(\mathbf{r}))^T \left(\phi^*_{h_i} - \phi^*_{\frac{h}{2}_i} \right) \\ & - \sum_{i=1}^{\mathcal{N}} \oint_{\Gamma_{IN}} d\Gamma \left(\phi^*_{h_i} - \phi^*_{\frac{h}{2}_i} \right) \mathbf{n} \cdot \mathbf{A} \left(\psi_h(\mathbf{r}) - \psi_{h_{IN}}(\mathbf{r}) \right). \end{aligned} \tag{7.32}$$

The residual in the above expression, which is angularly discrete but spatially continuous, can be expanded:

$$\mathscr{R}(\psi_h(\mathbf{r})) = \mathbf{A} \cdot \nabla \psi_h(\mathbf{r}) + \mathbf{H}\psi_h(\mathbf{r}) - \lambda(\psi_h(\mathbf{r}))\mathbf{B}\psi_h(\mathbf{r}). \tag{7.33}$$

Inserting the finite element approximation of $\psi_h(\mathbf{r})$ into the expanded residual equation above, that is to make the substitution $\psi_h(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r})\phi_{h_j}$, one ob-

tains:

$$\mathscr{R}(\psi_h(\mathbf{r})) = \mathbf{A}\cdot\nabla\left(\sum_{j=1}^{\mathcal{N}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right) + \mathbf{H}\left(\sum_{j=1}^{\mathcal{N}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right) - \lambda_h\mathbf{B}\left(\sum_{j=1}^{\mathcal{N}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right). \tag{7.34}$$

Inserting equation (7.34) into equation (7.32), and also inserting the finite element approximations $\psi_h(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r})\phi_{h_j}$ and $\psi_{h_{IN}}(\mathbf{r}) = \sum_j \mathbf{N}_j(\mathbf{r})\phi^{IN}_{h_j}$ into the surface integral in equation (7.32), one obtains:

$$\begin{aligned}\lambda_{\frac{h}{2}} - \lambda_h = \sum_{i=1}^{\mathcal{N}}\int_V dV\mathbf{N}_i &\left(\mathbf{A}\cdot\nabla\left(\sum_{j=1}^{\mathcal{N}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right) + \mathbf{H}\left(\sum_{j=1}^{\mathcal{N}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right)\right. \\ &\left.-\lambda_h\mathbf{B}\left(\sum_{j=1}^{\mathcal{M}}\mathbf{N}_j(\mathbf{r})\phi_{h_j}\right)\right)^T\left(\phi^*_{h_i}(\mathbf{r}) - \phi^*_{\frac{h}{2}_i}(\mathbf{r})\right) \\ -\sum_{i=1}^{\mathcal{N}}\oint_{\Gamma_{IN}} d\Gamma\left(\phi^*_{h_i}(\mathbf{r}) - \phi^*_{\frac{h}{2}_i}(\mathbf{r})\right)&\mathbf{n}\cdot\mathbf{A}\sum_{j=1}^{\mathcal{M}}\mathbf{N}_j(\mathbf{r})\left(\phi_{h_j} - \phi^{IN}_{h_j}\right).\end{aligned} \tag{7.35}$$

Writing equation (7.35) in matrix form:

$$\begin{aligned}\lambda_{\frac{h}{2}} - \lambda_h &= \left[\mathbf{C}_{\frac{h}{2}}\mathbf{\Phi}_h\right]^T\left(\mathbf{\Phi}^*_h - \mathbf{\Phi}^*_{\frac{h}{2}}\right) \\ &= \mathbf{R}_{\frac{h}{2}}(\mathbf{\Phi}_h)^T\left(\mathbf{\Phi}^*_h - \mathbf{\Phi}^*_{\frac{h}{2}}\right),\end{aligned} \tag{7.36}$$

in which $\mathbf{C}_{\frac{h}{2}}$ is a matrix containing the coefficients of the discrete transport equation on the $\mathcal{O}(\frac{h}{2})$ mesh. The element i,j of this matrix is defined as:

$$\begin{aligned}\mathbf{C}_{\frac{h}{2}_{ij}} = \int_{V_E} dV\,(&\mathbf{A}\cdot\mathbf{N}_i(\mathbf{r})\nabla\mathbf{N}_j(\mathbf{r}) + \mathbf{H}\mathbf{N}_i(\mathbf{r})\mathbf{N}_j(\mathbf{r}) \\ &- \lambda_h\mathbf{B}\mathbf{N}_i(\mathbf{r})\mathbf{N}_j(\mathbf{r})),\end{aligned} \tag{7.37}$$

where $\mathbf{N}_i(\mathbf{r})$ is the node i $\mathcal{O}(\frac{h}{2})$ finite element shape function at position $\mathbf{r}$. Note this matrix will contain the eigenvalue λ_h that is a function of the $\mathcal{O}(h)$ eigenvector. This completes the derivation of equation (7.31), and shows that it is equivalent to the defect equation in the continuum, equation (7.30).

7.10. Numerical Examples

In this section, numerical results obtained from the defect correction procedure in transport problems are discussed. A scattering problem and an absorber problem are considered. The performance of each enrichment scheme is shown. To validate the error recovery procedure has been implemented correctly in the code, the full quadratic eigenvector and the full cubic eigenvector were in turn placed into the defect equation, rather than using the enriched solution. When this was done, the defect equation was found to recover the full error in the eigenvalue. That is the eigenvalues obtained from it were those obtained from the inverse power iteration when those element types are used, rather than a linear element type. This validates both the theory behind the correction scheme and also that it has been implemented correctly. Both the continuum error recovery and the discrete error recovery achieve this. Figure (7.4) shows that the full cubic eigenvalue and the full quadratic eigenvalue are recovered in full when the full quadratic and full cubic eigenvectors are placed into the defect equation rather than the enriched solution.

7.10.1. Absorber Problem

The first problem to be considered is a two-dimensional absorber problem. This comprises a fissile material region in the range $\{0.6 \leq x \leq 1.0\}, \{0.6 \leq y \leq 1.0\}$. A scattering cross-section $\sigma_s = 0.0cm^{-1}$ is used throughout the domain and a total cross-section $\sigma_t = 10.0cm^{-1}$. The fission cross-section of the fissile material region is $\sigma_f = 10.0cm^{-1}$. Eigenvalue calculations were performed on this problem using a linear element type, that is a finite element shape function of degree $p = 1$. These were done using different mesh spatial resolutions. Due to the Galerkin orthogonality, an enriched eigenvector must be placed into the defect equation in order to obtain a correction to the eigenvalue. To provide this, the linear element eigenvector was enriched using each of the eigenvector enrichment schemes described in this chapter. The eigenvector enriched via these schemes was then placed in the defect equation. Results using discrete ordinates are compared with results that use spherical harmonics. These results are shown in figure (7.5). The uncorrected eigenvalue obtained using a linear element type is plotted. For benchmarking, $p = 2$ and $p = 3$ element calculations are included, without correction. Then $p = 1$ eigenvalue is then corrected using eigenvectors enriched with the local quadratic (LOCQ) method that was introduced in section (5.5); the

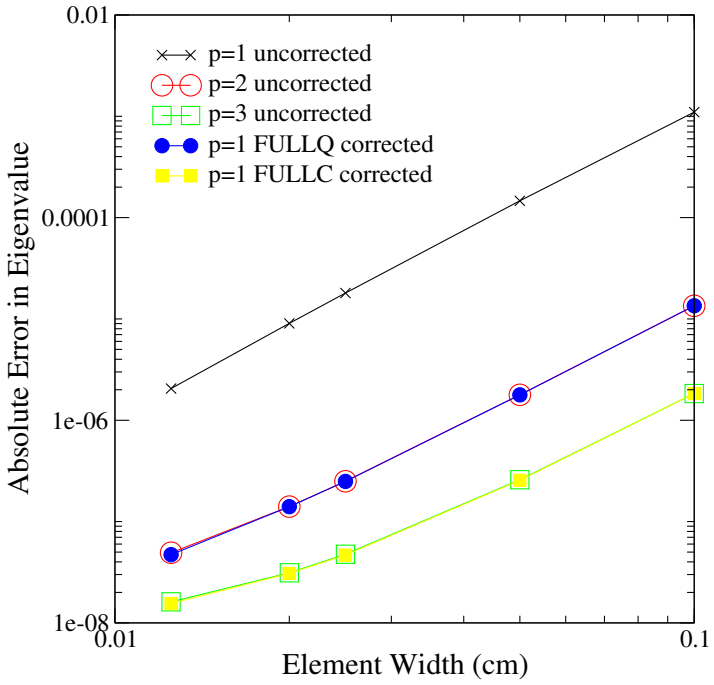
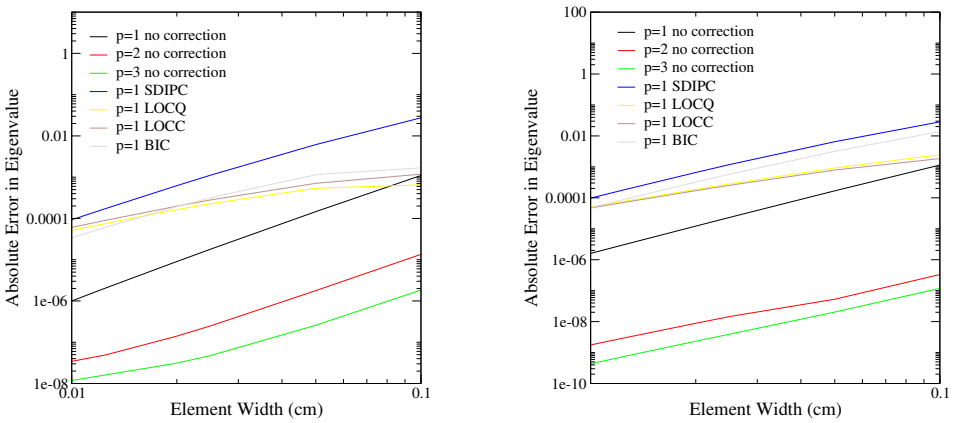


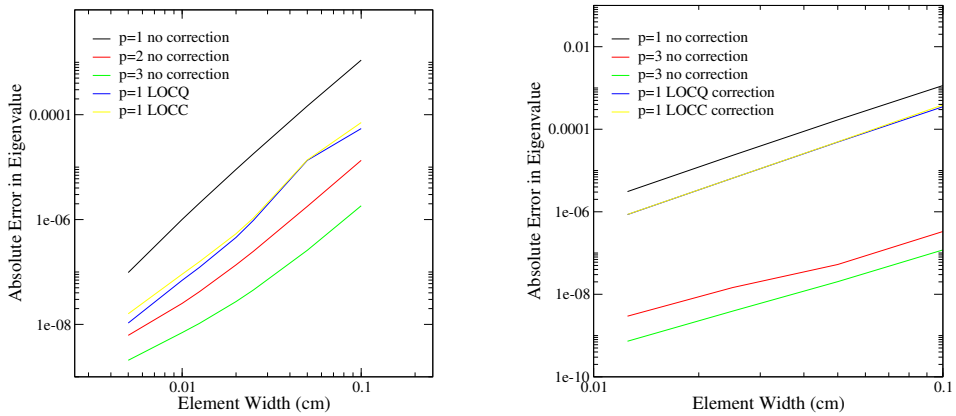
Figure 7.4.: The full error in the eigenvalue is recovered when the full quadratic eigenvector or the full cubic eigenvector is placed into the defect equation instead of the enriched solution.



(a) P_3 (b) S_6

Figure 7.5.: Eigenvalue error recovery in the absorber problem depends on reliable enrichment of the low-order eigenvector.

local cubic (LOCC) method, in which the quadratic element used in the LOCQ method is swapped with a cubic element; a cubic Serendipity element (SDIPC) and a bicubic surface fit (BIC). These schemes are designed to smooth a low-order eigenvector so it may be placed into the defect equation. This does not give an improvement in the absorber problem. However, allowing the local quadratic method and local cubic method to iterate on the fission source, some improvement is obtained. This result in shown in figure (7.6). However, only part of the error is



(a) P_3 (b) S_6

Figure 7.6.: Allowing the local quadratic (LOCQ)and the local cubic (LOCC) sweeps to iterate on the fission source provides a more reliable eigenvector enrichment that can lead to eigenvalue error recovery.

recovered via this approach. Allowing the local solutions to iterate on the fission source increases the computational overhead considerably, and is not necessarily a satisfactory approach.

7.10.2. Scattering Problem

The second problem to be considered is a scattering problem. This comprises a fissile material region placed inside a heavy scattering material. The fissile material is located in the range $\{0.5 \leq x \leq 1.0\}, \{0.5 \leq y \leq 1.0\}$ and has a fission cross-section of $\sigma_f = 1.0cm^{-1}$. A total cross-section of $\sigma_t = 1.0cm^{-1}$ is used throughout the domain. A scattering cross-section of $\sigma_s = 0.9cm^{-1}$ is used in both the fissile region and in the absorber. As with the previous problem, eigenvalues on a $p = 1$ element type are obtained via inverse power iteration at various spatial mesh resolutions. The linear element eigenvectors are then enriched us-

ing the eigenvector enrichment schemes described in this chapter. The enriched eigenvectors are then placed in the defect equation. Results, for each eigenvector enrichment scheme, are shown in figure (7.7). As with the absorber problem,

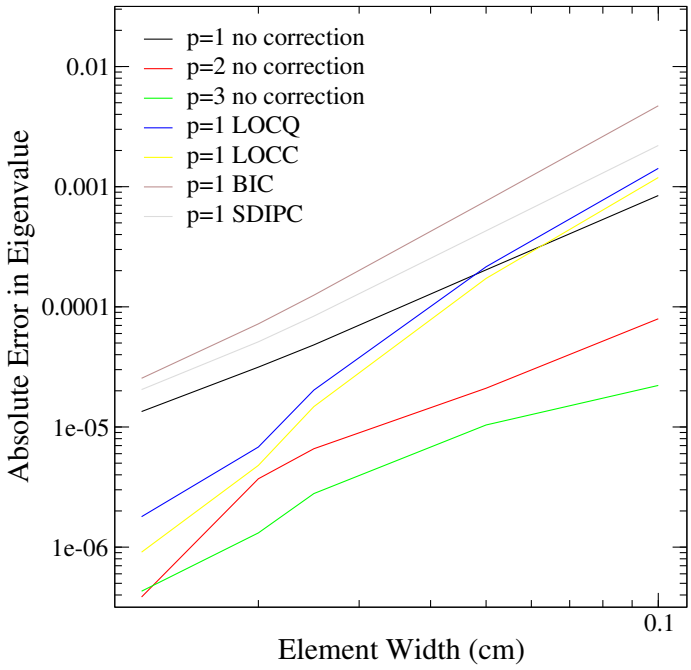


Figure 7.7.: Performance of eigenvalue enrichment schemes for eigenvalue error recovery in a scattering problem.

uncorrected results are included for $p = 1, p = 2, p = 3$ element types. The higher-order element type are included for benchmarking. It is seen from these results that the eigenvalue error recovery depends on a more reliable eigenvector enrichment scheme in transport problems. This is recommended as the topic of further work.

7.11. Conclusions

Corrections to the eigenvalue in DG transport calculations have been derived. Computation of the defect in the eigenvalue requires the low-order solution to be enriched. The defect recovery equations indicate how to improve the eigenvalue in transport problems, given this enriched eigenvector. However, they do not indicate how the low-order eigenvector is to be enriched. Different strategies

for obtaining the error, using several enrichment schemes, have been demonstrated on two dimensional geometry on a regular orthogonal grid. Enrichment schemes described include a bicubic fit to the discontinuous data internal to an element, which uses interpolating splines that take advantage of the orthogonal grid strategy used in the present work; fitting the low-order solution to a cubic Serendipity element which avoids the need for cross-derivatives to be calculated; and a third scheme that decomposes the discontinuous low-order solution into a continuous and a discontinuous component, and adds local smoothness to the continuous part. Fitting the low-order solution to a higher-order element, and allowing an iteration on the fission term, leads to improved estimates of the system eigenvalue, and an improvement in convergence rate. However, correction is not as sharp and not as reliable as the correction obtained when an elliptic operator is used to obtain the eigenvector in smooth diffusive problems. It is suggested this is due to the shape of the operators, and hyperbolic operators are sensitive to nearby shocks and the resulting eigenvector is naturally unsmooth, whereas elliptic operators being sensitive to the boundary of the domain are internally very smooth. This could be investigated further with use of the maximum principle and mean value theorem.

CHAPTER 8

CONCLUSION

Synopsis

This Thesis has presented some new discretisation methods based on discontinuous Galerkin approaches that add robustness and numerical accuracy to the finite element discretisation of the Boltzmann Transport Equation. It has been demonstrated how these are applied in computational radiation transport modelling. Adjoint-based error recovery schemes for functional and eigenvalue improvement have also been introduced. This final chapter reflects on this work and brings together the conclusions of each section. The work that has been completed is well aligned with the current research interests of AWE and Imperial College, and so a future work plan has been included to help ensure that the research presented in this Thesis is continued.

Contents

8.1. Summary

The scope of this Thesis has been outlined in chapter (1). The Thesis was a collaborative programme of research between AWE plc and Imperial College, London. The aim of the research was to develop novel discretisations of the Boltzmann Transport Equation that are based on discontinuous Galerkin formulations. The new schemes included linear and non-linear stabilisation strategies that were incorporated in the discretisation of the Boltzmann Transport Equation. Separate schemes for removing defects in functional problems and eigenvalue problems were developed. These used the adjoint system of equations to derive *a posteriori* methods for removing errors (defects) from the functional and eigenvalue solutions in transport problems. The need to enhance the robustness of existing numerical schemes in the context of computational radiation transport modelling has been discussed. A work plan was presented that illustrated the research to be done during this Thesis. A brief review of the literature relating to the relevant methods was given. The Thesis is concluded with some suggestions for future work.

8.2. DG Methods for The Boltzmann Transport Equation

Some background to the present work was given in chapter (2), which was followed by an introduction of the one-speed linear form of the time-dependent Boltzmann Transport Equation. Three of the most practical methods for discretising this equation in angle were discussed, and a very general procedure described for arriving at the angular discrete form. The spatial and temporal discretisation of the angular discrete form was then introduced. This was done using the Riemann

finite element method. Some prototype methods were then developed in one- dimension, that enhance the basic discontinuous Galerkin scheme. These included the Upwind Average approach, where additional upwinding in the boundary of the element is used; two Petrov-Galerkin methods that differ in how the stabilisation on the boundary between elements is treated and the use of sub-grid-scale bubble functions in the discretisation. The Within-Element Petrov-Galerkin method was selected for further development because is it was a good compromise between accuracy and lack of complexity of implementation, and also because of the scope that it had for further development. This method became the topic of the chapters that followed.

8.3. Stabilisation

A Petrov-Galerkin method was selected from a variety of discontinuous methods that had previously been demonstrated in one-dimensional problems in chapter (2). Chapter (3) developed this method further to form a new stabilising discretisation scheme. The concept of upwind stabilisation was introduced. It was shown that one improves the stability of the discontinuous Galerkin method by adding artificial dissipation to the discretisation. However, this is only successful if one avoids adding too much dissipation. If too much dissipation is added, more particles can be advected backwards across an element than there are entering the element. This is very detrimental to the discretisation and to avoid it, upwinding needs to be controlled and if necessary, the magnitude of the upwind term needs to be limited. The chapter extended the method by developing an optimal coefficient based on material properties. The purpose of the optimal coefficient was to control the magnitude of the upwind term so that a suitable amount of artificial dissipation was used. Two types of upwind scheme were considered, both of which used an optimal coefficient. One was a Petrov-Galerkin method that added the dissipation internal to an element and the other added on a diffusion term. The two schemes required a different optimal coefficient definition. In one- dimensional problems, both schemes were shown match the analytical solution precisely, at the outlet node of each element; the optimal coefficient tuned the amount of dissipation that needed to be added to the discretisation to achieve this. The scheme was extended to multi-dimensional problems, in which it consistently performed, removing unwanted oscillation from the transport solution on a range of problems using different types of angular discretisation. However, a major drawback of these

optimal methods was that they used the same magnitude of dissipation throughout a given material. Another issue they had was their linearity; the added dissipation was only able to act in the streamline direction. Chapter (3) concluded that the optimal schemes could be improved by respecting the fact that oscillations can occur in directions other than the streamline such as in the solution gradient direction, and that they should be enhanced to cope with this by adding non-linearity to the upwind term.

8.4. Non-Linearity

Non-linearity was added in chapter (4) to the optimal schemes developed in chapter (3). Non-linearity allowed the dissipation to be tuned locally for each element, based on the local behaviour of the solution. It addressed the issue of oscillations that occur in the gradient direction as well as those that occur in the streamline direction. This was formulated by projecting the solution gradient onto the streamline, using the cosine rule. The length of this projection was then used as the length scale in the stabilisation term. Essentially, this controlled how far into an element the artificial dissipation acted. Although the dissipation still acted in the streamline direction, the new choice of length scale optimised it in the direction of the solution gradient. The optimal linear methods of chapter (3) were shown to work well in steady-state problems. Difficulties were encountered, however, when using them in time-dependent problems. One of the issues was the choice of length scale in the linear scheme; in time-dependent problems it was not originally clear how this should be defined. For example, the element width was a good choice of length scale where there was steady-state behaviour (at late time, for example) but was not suitable in the early stages of a calculation where there was transient behaviour, at which point the element time-step would have been a better choice. Chapter (4) concluded that using non-linearity in the stabilisation scheme addressed this issue, by defining locally the most appropriate amount of dissipation to use. Results demonstrated the non-linear scheme performed consistently in a variety of problems that included time-dependent and steady-state cases. It was suggested that goal-based approaches should be investigated besides the approaches discussed so far, the aim being that functionals and eigenvalues may be improved.

8.5. Functional Defect Improvement

The first of these goal based approaches were introduced in chapter (5), initially with the emphasis on improving linear functionals that arise in transport applications, using an *a posteriori* method for error measure. Linear functionals that occur in radiation transport problems include power in reactor fuel pins, radiation shielding input flux and source-detector problems. The error measure that was developed for this class of problem used the adjoint system of equations to obtain the sensitivities a particular functional has. The adjoint equations were solved inline with the forward problem. One then convolved the adjoint eigenvector with the residual of the forward problem to produce an approximation to the error in the functional. This procedure was derived from first order Taylor analysis. The approximation to the error was then subtracted from the functional. Results were presented that showed this method to dramatically improve the convergence rate of the functional, with respect to mesh resolution. An approach was outlined that one might use to correct second order functionals; this was based on second order Taylor analysis. For source detector problems, however, a linear response was considered more representative. Chapter (5) concluded by recommending adjoint-based methods for defect recovery in functionals, and that it was worth investigating their potential to form error metrics to automate mesh adaption. However, the conclusion stated that any work on mesh adaption was beyond the scope of this Thesis, although extremely relevant, and suggested instead the application of the method to eigenvalue improvement.

8.6. Eigenvalue Defect Improvement

The second goal-based approach was introduced in chapter (6), and extended the functional correction procedure to the K_{eff} eigenvalue in criticality problems, as suggested. This was done initially for elliptic operators and so was demonstrated in diffusion problems. It was shown that first order Taylor analysis of the residual and eigenvalue are sufficient if certain normalisations such as fission normalisation, are applied to the eigenvector. The adjoint equations were applied in a similar manner to the way they were used in the previous chapter, but to the K_{eff} eigenvalue. It was shown that the adjoint equations had to be discretised differently to the forward equations. This was due to the Galerkin orthogonality, which otherwise resulted in zero correction. The remedy for this was to enrich the ad-

joint eigenvector that feeds into the defect equation. This enrichment was done using a bicubic smoothing, which approximates a higher order solution and could be done in a post-process step. First the forward model was solved using a conventional inverse power iteration. The adjoint system was then solved in-line. The eigenvectors were then normalised to the fission source. A defect iteration was then applied to the eigenvalue in which an approximation to the error was derived and subsequently removed. Results were shown to improve the convergence rate of K_{eff} with respect to mesh resolution. The K_{eff} correction procedure was then applied to hyperbolic operators in chapter (7) to demonstrate the use of this theory in transport problems. The operator in transport problems was shown to present eigenvalues that are difficult to correct. It was suggested this was because the eigenvector solutions were difficult to enrich; a requirement needed in order to use the correction procedure. A number of methods were then presented that were designed to perform this enrichment in transport problems so that reliable eigenvalue corrections could be applied. It was suggested that although the defect correction scheme was able to indicate how to improve the eigenvalue in transport problems, given an improved eigenvector, it was not able to indicate how to improve the eigenvector in order to do this. Fitting the low-order eigenvector solution to a higher-order element type, and allowing an iteration on the fission source led to an improvement in the eigenvalue in a transport problem. However, correction was not as sharp, and not as smooth as that obtained from elliptic operators in smooth continuous problems. For elliptic operators, it was suggested, solutions are naturally smooth, so a bicubic surface fit was usually successful in enriching the solution while for hyperbolic operators, the solution is not necessarily any smoother than the boundary condition due to the shock-capture characteristics of the operator. Consequently, chapter (7) concluded, smoothing the eigenvector in transport problems does not always improve it, and therefore the success of the defect recovery scheme in transport problems depends on how reliable the eigenvector enrichment method is that one uses on the low-order adjoint eigenvector solution.

8.7. Future Work

In this section, a plan of future work is suggested to help to ensure the research presented in the Thesis is continued. Almost every area of the Thesis could be extended. Ideally this would be in collaboration with other students working on

areas that relate to the items discussed. Below are some suggestions.

8.7.1. Cylindrical Coordinates

In the present work, consideration has only been given to Cartesian geometries. It is suggested that the Petrov-Galerkin methods presented in chapters (3) and (4) be implemented in two-dimensional cylindrical (R-z) coordinates. This should present a straight forward piece of work, and is believed to be the first time any type of Petrov-Galerkin method has been applied within a DG framework in coordinates. This would expand, quite considerably, the range of problems and applications of the methods. Use of standard Petrov-Galerkin methods, such as those in *RADIANT* , in axisymmetry combined with DG is also of interest, and believed to be novel work.

8.7.2. Multi-grid Solution Strategy

A great deal of effort has been afforded to the development of multi-grid solver technology at Imperial College (Buchan, 2007). This makes large scale radiation transport calculations feasible on unstructured computational meshes. Implementation of the schemes in this Thesis on unstructured meshes are an obvious extension to the present work. This would require a more sophisticated solution scheme than that in the codes used to run the calculations in this Thesis, which take advantage of structured orthogonal meshes. This is a more complicated piece of work than extension to axisymmetry.

8.7.3. Sub-grid Development

Sub-grid schemes have advanced considerably in the last few years (Candy, 2010), and their relevance to radiation transport demonstrated (Buchan, 2005b). In the present work, including sub-grid quadratic bubble functions in the finite element space, to give the discretisation a multi-scale properties is of a great deal of interest. It is suggested this be tried in multi-dimensional problems. These scheme naturally stabilise the element and relate quite closely with Petrov-Galerkin and SUPG schemes. They would also be useful for benchmarking the stabilisation schemes presented in this Thesis. A corresponding non-linear optimal method could be developed for sub-grid technologies. The advantages are fewer unknowns than standard discontinuous methods, but loss of accuracy. Developing an optimal non-linear method could recover some of this accuracy.

8.7.4. Parallelisation

Full parallelisation of the methods presented would be useful, this would assist with running larger problems. Particularly for the eigenvalue correction schemes. To benchmark the error correction procedures, it was necessary to run very substantial problems on fine computational meshes. In order to reduce the eigenvalue errors to obtain a benchmark solution, resolutions had to be used that were very difficult to achieve with a serial code. If more work on the eigenvalue correction schemes is to be done, then a parallel code should be developed for this purpose. The eigenvalue correction scheme requires in-line calculation of the adjoint. This typically doubles the overhead of the scheme, so run-time reduction offered by a multi-process implementation would be especially useful. Parallelisation is essential if any of the schemes are to be developed for three-dimensional time-dependent problems.

Choice of Solver

If parallelisation of the method is pursued, a decision would need to be made whether to use a sweep-based solver such as that already implemented in the code, or whether to use for example a multi-grid approach, as this would have very strong implications for extracting parallelism. It is perhaps more obvious how to extract parallelism from a sweep-based approach than it is from a multi-grid approach. In this case, the method of spatial domain decomposition is well proven and its implementation relatively straight forward. However, a sweep-based approach may not be the most efficient method for some problems, for example P_N and LW_N on unstructured grids. Clearly, the merits of different solvers would need to researched before a sensible strategy could be chosen.

8.7.5. Defect Correction

It is possible the defect correction schemes that were applied to source-detector problems and eigenvalues could be extended in several ways that could form the basis of an interesting, and very useful project. One of these ways in which they might be developed is to extend either a one-dimensional or a two-dimensional model to a three dimensional domain without the three dimensional problem having to be solved. This is achievable by ”buckling” one of the dimensions of the problem. Another potential avenue to explore is what other applications may exist

for the adjoint-based method; wherever there is use for the sensitivity map of a problem, the scheme developed in this Thesis may be applicable. For example, to automate mesh adaption by defining where in the phase space of the equations the errors are large. Also, it could be investigated how useful these methods are for angular refinement. Their ability to work in different angular frameworks has been demonstrated so it should be possible to use them to define suitable metric for refining (or de-refining) the angular bases. This could be a significant piece of work; the requirements of angular refinement differ from those of spatial refinement.

8.7.6. High-Order Element Types

The correction schemes developed in chapters (5) and (6) were only applied to linear element types. Results from higher order element types (quadratic and cubic bases) that did not use the correction procedure, were included for benchmarking the error recovery process. It was seen in most problems that the higher the order of the element the faster the convergence rate was. By plotting the absolute error against the number of elements, one can use the slope to determine these convergence rates. One will find that in many problems, the slopes are steeper for higher order elements. Perhaps a goal-based technique could be developed that automates p refinement within the DG framework. This would likely be very successful where the error slopes are steeper for the higher order elements. In problems where the convergence is similar for different order elements, AMR would perhaps be a more successful strategy. The defect scheme could perhaps be used to chose between the two, so they may be combined into one code. Increasing the order of the element also improves computational efficiency, due to improved memory accessing and avoiding the in-direct addressing that can make AMR schemes become inefficient. This is especially true on unstructured meshes.

8.7.7. Optimal Petrov-Galerkin with Higher-Order Coefficient

The optimal Petrov-Galerkin methods that were developed in chapter (3) used an optimal coefficient that was based on one-dimensional considerations. The expression that was derived for this optimal coefficient used a one-dimensional finite element projection of the problem equated to the analytical solution. This finite element projection assumed a linear element type. It would be interesting to repeat the derivation of these expressions for the optimal coefficient using higher order projections. This should be a relatively straight forward extension to the present

Petrov-Galerkin formulations.

8.7.8. Petrov-Galerkin in Angle as Ray-Effect Palliative

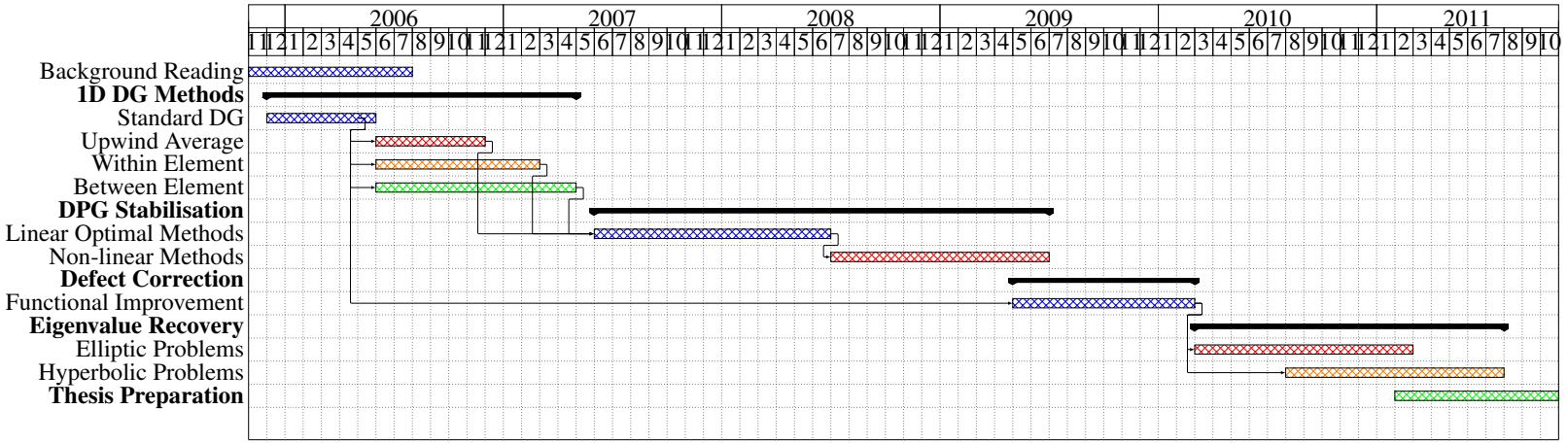
In the present work, Petrov-Galerkin weighting has been applied in the spatial discretisation to enhance calculations and improve behaviour of the solution. The benefits of this approach are shown in chapters (3) and (4), in particular for the control of unwanted oscillation on the transport solution. It is suggested as a future topic, to assess whether Petrov-Galerkin weighting can reduce unwanted oscillation in angle. This may smooth defects in the solution that are due to the angular approximation, for example the ray-effect present in all S_N calculations. Applying additional upwinding in angle may achieve this by introducing coupling between the rays, with the effect of diffusing particles between angles. If successful, this would provide a ray-effect palliative that could improve angular convergence.

Acknowledgements

I would like to thank both my supervisors, Professor Christopher Pain and Professor Richard Smedley-Stevenson for their technical guidance, their support and their patience throughout my PhD. I have learnt a great deal through working with them over the last six years on this PhD. I acknowledge their management and planning experience, and their technical knowledge that made this Thesis possible. I must also acknowledge the financial support of AWE plc, and also the Computational Physics Group whom funded this work. It has been an extremely successful collaboration between the Applied Modelling and Computation Group at Imperial, and The Computational Physics Group at AWE. It has established links with researchers both at Imperial College and elsewhere in academia. I thank AWE plc for giving me the opportunity to present my work regularly at large international conferences such as the M&C, PHYSOR and ICTT. This has undoubtedly strengthened my work and helped to gain visibility in my research area. I thank AWE plc for the time they gave me to complete this research, a significant part of which used AWE's time. I wish to thank Doctor Stephen Hughes and Doctor Michael Sleigh for the flexibility they gave me. I acknowledge the technical advice given to me during the course of the work from a number of colleagues, in addition to that given by my two supervisors. For this I thank Doctor Andrew Buchan, Doctor Christopher Baker, Doctor Mathew Eaton, Doctor David Ham (Imperial College) and David Barrett (AWE plc). Finally I thank my friends and family.

APPENDIX A

WORK PLAN



APPENDIX B

SOURCE ITERATION PROCEDURE

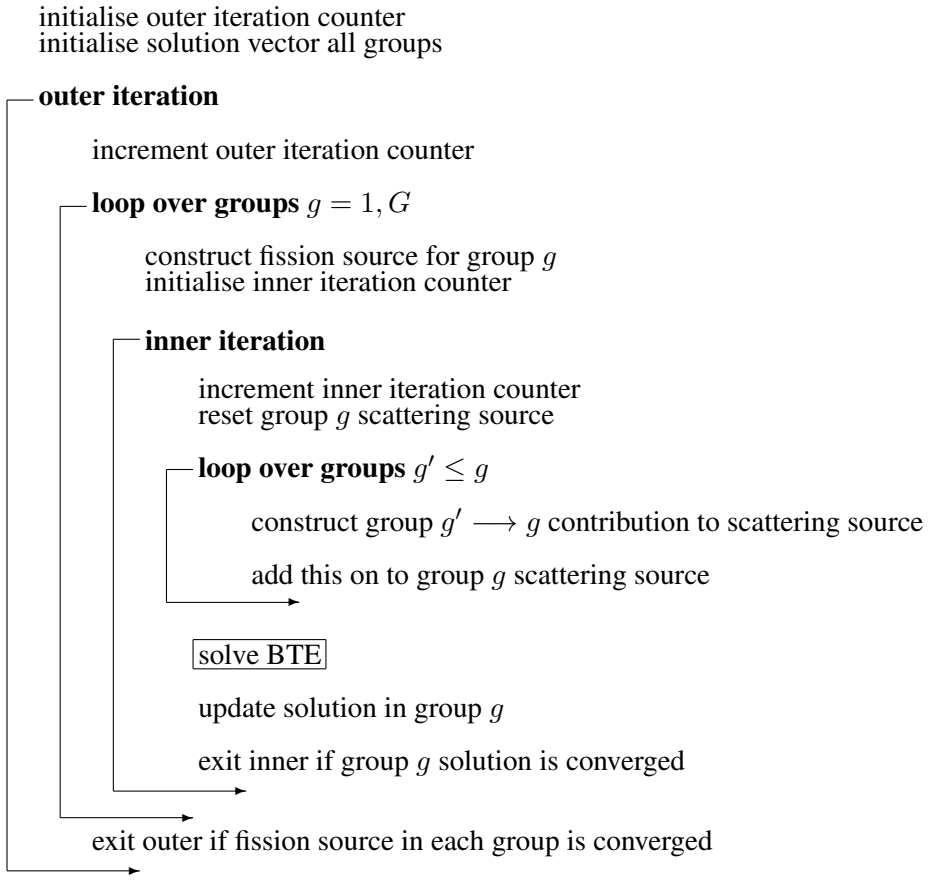


Figure B.1.: Source Iteration scheme used in many transport codes.

APPENDIX C

ADJOINT OPERATOR DERIVATION

C.1 Adjoint Definition

The adjoint operator $\mathscr{L}^*$ is defined in the following identity of the inner products:

$$\langle \psi^*, \mathscr{L}\psi \rangle = \langle \psi, \mathscr{L}^*\psi^* \rangle \,, \tag{C.1}$$

in which ψ and ψ^* are the forward and adjoint eigenvectors, respectively and $\mathscr{L}$ the forward operator. This identity is taken from (Lewis and Miller, 1993, page 48). In radiation transport problems the operator $\mathscr{L}$ acting on the eigenvector at position $\mathbf{r}$, travelling in direction Ω at energy E forms the set of forward equations which can be written:

$$\begin{aligned} \mathscr{L}\psi(\mathbf{r},\Omega,E) &= \Omega\cdot\nabla\psi(\mathbf{r},\Omega,E) + \sigma_t(\mathbf{r},E)\psi(\mathbf{r},\Omega,E) \\ &- \int_{\Omega'} d\Omega' \int_{E'} dE' \sigma_s\left(\mathbf{r},\Omega'\rightarrow\Omega,E'\rightarrow E\right)\psi(\mathbf{r},\Omega',E'), \end{aligned} \tag{C.2}$$

in which $\sigma_t(\mathbf{r},E)$ is the total cross-section at position $\mathbf{r}$ for particles with energy E and $\sigma_s(\mathbf{r},\Omega'\rightarrow\Omega,E'\rightarrow E)$ the cross-section of the scattering of particles from direction Ω' into direction Ω and from energy E' into energy E. The adjoint eigenvector $\psi^*(\mathbf{r},\Omega,E)$ contains information about the first-order sensitivities in the forward eigenvector $\psi(\mathbf{r},\Omega,E)$. One may therefore weight the right hand side of the above equation with the adjoint eigenvector instead of a finite element basis function, in order to derive approximations to errors in the forward set of equations.

This error can subsequently be used to improve initial estimates of functionals or multiplication factors. The error can also be used to automate mesh adaption or expedite power iteration convergence. Note that the operation of $\mathscr{L}$ on $\psi(\mathbf{r}, \Omega, E)$ constructs the residual $\mathscr{R} = \mathscr{R}(\psi)$ of the transport equation, which may therefore be written:

$$\mathscr{L}\psi(\mathbf{r}, \Omega, E) = \mathscr{R}(\psi(\mathbf{r}, \Omega, E)). \tag{C.3}$$

Note that the residual $\mathscr{R}$ is a function of the eigenvector $\psi(\mathbf{r}, \Omega, E)$.

C.2 Operator Derivation

Expressing the inner product on the left hand side of equation (C.1) as an integral, which is from the definition of the dot product, one obtains:

$$\begin{aligned}\langle \psi^*, \mathscr{L}\psi \rangle = &\int_V dV \psi^*(\mathbf{r}, \Omega, E) \left(\Omega \cdot \nabla\psi(\mathbf{r}, \Omega, E) + \sigma_t \psi(\mathbf{r}, \Omega, E)\right) \\ &- \int_{\Omega'} d\Omega' \int_{E'} dE' \sigma_s \left(\mathbf{r}, \Omega' \to \Omega, E' \to E\right) \psi(\mathbf{r}, \Omega', E').\end{aligned} \tag{C.4}$$

Applying the divergence theorem to the advection term in the above equation, in exactly the same way as done when the finite element basis functions are used instead of the adjoint eigenvector in the weighting of the equation, one can write:

$$\begin{aligned}\langle \psi^*, \mathscr{L}\psi \rangle = &\int_V dV \left(-\nabla\psi^* \cdot \Omega\psi + \psi^* \sigma_t \psi\right) + \oint_\Gamma d\Gamma \mathbf{n} \cdot \Omega \psi^* \psi \\ &- \int_V dV \int_{\Omega'} d\Omega' \int_{E'} dE' \psi^* \sigma_s \left(\mathbf{r}, \Omega' \to \Omega, E' \to E\right) \psi,\end{aligned} \tag{C.5}$$

in which the dependence of ψ and ψ^* on $(\mathbf{r}, \Omega, E)^T$ has been assumed, and shall be assumed from this point on. Using the identity $\psi\Omega \cdot \nabla\psi^* = \nabla\psi^* \cdot \Omega\psi$, and the condition on the boundary $\psi^*(\mathbf{r}, \Omega, E) = 0, \mathbf{r} \in \Gamma, \mathbf{n} \cdot \Omega \geq 0$ that specifies no

adjoint flux enters the domain, one can write the above as:

$$\langle \psi^*, \mathscr{L}\psi \rangle = \int_V dV \left(-\psi\Omega \cdot \nabla\psi^* + \psi^*\sigma_t\psi \right)$$
$$- \int_V dV \int_{\Omega'} d\Omega' \int_{E'} dE' \psi^* \sigma_s \left(\mathbf{r}, \Omega' \to \Omega, E' \to E \right) \psi. \tag{C.6}$$

Using the fact that $\psi^*\sigma_s\psi = \psi\sigma_s\psi^*$ while interchanging E and E' in the integral over energy and interchanging Ω and Ω' in the integral over angle in the scatter term, one can write:

$$\langle \psi^*, \mathscr{L}\psi \rangle = \int_V dV \left(-\psi\Omega \cdot \nabla\psi^* + \psi\sigma_t\psi^* \right)$$
$$- \int_V dV \int_{\Omega} d\Omega \int_{E} dE \psi \sigma_s \left(\mathbf{r}, \Omega \to \Omega', E \to E' \right) \psi^*. \tag{C.7}$$

The adjoint operator $\mathscr{L}^*$ acting on the adjoint eigenvector ψ^* at position $\mathbf{r}$, travelling in direction Ω' at energy E' forms the set of adjoint equations which can be written:

$$\mathscr{L}^*\psi^*(\mathbf{r}, \Omega', E') = -\Omega' \cdot \nabla\psi^*(\mathbf{r}, \Omega', E') + \sigma_t\psi^*(\mathbf{r}, \Omega', E')$$
$$- \int_{\Omega} d\Omega \int_{E} dE \sigma_s \left(\mathbf{r}, \Omega \to \Omega', E \to E' \right) \psi^*(\mathbf{r}, \Omega', E'). \tag{C.8}$$

Note the above expression is the same as equation (C.2) for the forward model but with Ω changed to $-\Omega'$ in the advection term and E changed to E' and Ω changed to Ω' in the scatter term. Thus in the adjoint system, information travels in the opposite direction in space, angle and energy (and time if this is included in the phase space) with respect to the forward model. Substituting equation (C.8) into equation (C.7) one obtains:

$$\langle \psi^*, \mathscr{L}\psi \rangle = \int_V dV \psi \mathscr{L}^*(\psi^*) = \langle \psi, \mathscr{L}^*\psi^* \rangle, \tag{C.9}$$

which is the same as equation (C.1)

...I finish knowing a great deal about very little.

References

Ackroyd, R. T., 1997. Finite Element Methods for Particle Transport. John Wiley & Sons Limited.

Adams, M. L., 1998. Discontinuous Finite Element Solutions of Slab-Geometry Discrete Ordinates Equations. In: Transactions of the 1998 Winter Meeting, Washington, DC.

Ainsworth, M., Oden, J. T., 1997. A Posteriori Error Estimation in Finite Element Analysis. Computer Methods in Applied Mechanics and Engineering 142, 1–88.

Alcouffe, R. E., 2003. A Robust Linear Discontinuous Method For The RZ, S_N Transport Equation. Transactions - American Nuclear Society ISSU 89, 363–366.

Alcouffe, R. E., Larsen, E. W., W. F. Miller, J., Wienke, B. R., 1979. Computational Efficiency of Numerical Methods for the Multigroup Discrete Ordinate Neutron Transport Equation in the Slab Geometry Case. Nulcear Science And Engineering 71, 111–127.

Andersen, F. M. B., 1997. Marshak Boundary Condition Recurrence Formulae. Z. angew. Math. Phys. 48, 165–170.

Azmy, Y., Sartori, E., 2010. Nuclear Computational Science. Springer.

Baker, C. M. J., 2011. Anisotropic Adaptivity and Subgrid Scale Modelling for the Solution of the Neutron Transport Equation with an emphasis on Shielding Applications. Ph.D. thesis, Imperial College London, Dept. Earth Sci. and Eng.

Beau, G. J. L., Tezduyar, T. E., 1991. Finite Element Computation of Compressible Flows with the SUPG Formulation. Advances in Finite Element Advances in Fluid Dynamics 123, 21–27.

Borgers, C., Larson, E. W., Adams, M. L., 1992. The Asymptotic Diffusion Limit of a Linear Discontinuous Discretization of a Two-Dimensional Linear Transport Equation. J. Comp. Phys. 98, 285–300.

Brezzi, F., Franca, L. P., Russo, A., 1998. Further Considerations on Residual-Free Bubbles for Advective-Diffusion Equations. Computational Methods in Applied Mechanics and Engineering.

Briggs, L. L., Miller, W. F., 1975. Ray-Effect Mitigation in Discrete Ordinate-Like Angular Finite Element Approximations in Neutron Transport. Nulcear Science And Engineering 57, 205.

Brooks, A. N., Hughes, T. J. R., 1982. Streamline Upwind Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations. Comp. Meth. Appl. Mech. 32, 199.

Buchan, A. G., 2003a. Angular discretisation of the first order boltzmann transport equation. part 1 s_N and walsh functions. Tech. rep., Imperial College London, Dep. Earth Sci. Eng.

Buchan, A. G., 2003b. Angular discretisation of the first order boltzmann transport equation. part 2 linear spherical wavelets. Tech. rep., Imperial College London, Dep. Earth Sci. Eng.

Buchan, A. G., 2003c. Angular discretisation of the first order boltzmann transport equation. part 3 quadratic spherical wavelets. Tech. rep., Imperial College London, Dep. Earth Sci. Eng.

Buchan, A. G., 2005a. Linear and quadratic octahedral wavelets on the sphere for angular discrtetisations of the boltzmann transport equation. Annals of Nuclear Energy 32, 1224–1273.

Buchan, A. G., 2005b. Linear and Quadratic Octahedral Wavelets on the Sphere for Angular Discrtetisations of the Boltzmann Transport Equation. Annals of Nuclear Energy 32, 1224–1273.

Buchan, A. G., 2007. Self Adaptive Wavelets for the Angular Discretisation of the Boltzmann Transport Equation. Ph.D. thesis, Imperial College London, Dept. Earth Sci. and Eng.

Buchan, A. G., Candy, A. C., Merton, S. R., Pain, C. C., Eaton, M. D., Hadi, J., Goddard, A. J. H., Smedley-Stevenson, R. P., 2010. The Inner Element Sub-Grid Scale Finite Element Method for the Boltzmann Transport Equation. NUclear Science and ENgineering 164, 105–121.

Buchan, A. G., Merton, S. R., Pain, C., Smedley-Stevenson, R., 2011. Riemann Boundary Conditions for the Boltzmann Transport Equation using Arbitrary Angular Approximations. Annals of Nuclear Energy 38, 1186–1195.

Caldiera, A. D., 2005. Application of the P_N Method to the Matrix Form Transport Equation. Annals of Nuclear Energy 32, 1889–1894.

Candy, A., 2010. Subgrid Scale Modelling of Transport Processes. Ph.D. thesis, Imperial College London, Dept. Earth Sci. and Eng.

Cheng, Y., Shu, C.-W., 2007. A discontinuous galerkin finite element method for time dependent partial differential equations with higher order derivatives. Math. Comp. S 0025-5718(07)02045-5.

Davison, B., 1957. Neutron Transport Theory. Oxford University Press, Ch. 10.

Delfour, M., Trochu, F., 1978. Discontinuous Galerkin Methods for the Approximation of Optimal Control Problems Governed by Hereditary Differential Systems. Springer Verlag, 256–271.

Donea, J., Huerta, A., 2003. Finite element methods for flow problems. John Wiley & Sons Inc. 33, 704–710.

Eaton, M. D., 2004. A High-Resolution Riemann Method for Solving Radiation Transport Problems on Unstructured Meshes. Ph.D. thesis, Imperial College London, Dept. Earth Sci. and Eng.

Eaton, M. D., Pain, C. C., Bowsher, J., Smedley-Stevenson, R. P., de Oliveira, C. R. E., Goddard, A. J. H., Umpleby, A. P., 2003. Unstructured Finite Element Based Riemann Solvers for Time-Dependent and Steady-State Radiation Transport. Transport Theory and Statistical Physics 32(6-7), 693.

et al, E. A., 2005. Inversion of Nuclear Well-Logging Data using Neural Networks. Geophys Prospect ISSN: 0016-8025 53, 103–120.

Fletcher, J. K., 1983. A Solution of the Neutron Transport Equation using Spherical Harmonics. J. Phys. A: Math. Gen. 16, 2827–2835.

Fortin, M., Fortin, A., 1989. New Approach for the Finite Element Simulation of Viscoelastic Flows. Journal of Non-Newtonian Fluid Mechanics 32, 295–310.

Giles, M. B., Pierce, N. A., Suli, E., 2004. Progress in Adjoint Error Correction for Integral Fucntionals. Computing and Visualization in Science 6, 113–121.

Hill, T. R., 1975. ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation. Tech. rep., Los Alamos Scientific Laboratory LA-5990-MS.

Holloway, J. P., Brunner, T. A., 2005. Two-Dimensional Time Dependent Riemann Solvers for Neutron Transport. J. Comp. Phys. 210, 386–399.

Hughes, T. J. R., Franca, L. P., Mallet, M., 1987. A New Finite Element Formulation for Computational Fluid Dynamics: VI. Convergence Analysis of the Generalised SUPG Formulation for Linear Time-Dependent Multidimensional Advective-Diffusive Systems. Computer Methods in Apllied Mechanics and Engineering. 63, 97–112.

Hughes, T. J. R., Mallet, M., Mizukami, A., 1985. A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalised Streamline Operator for Multidimensional Advective-Diffusive Systems. Computer Methods in Apllied Mechanics and Engineering. 58, 305–328.

Hughes, T. J. R., Mallet, M., Mizukami, A., 1986. A New Finite Element Formulation for Computational Fluid Dynamics: II. Beyond SUPG. Computer Methods in Apllied Mechanics and Engineering. 54, 341–355.

Hughes, T. J. R., Tezduyar, T. E., 1984. Finite Element Methods for First-Order Hyperbolic Systems with Particular Emphasis on the Compressible Euler Equations. Computer Methods in Applied Mechanics and Engineering. 45, 217–284.

Hulbert, Hughes, 1990. Space-Time Finite Element Methods for Second-Order Hyperbolic Equations. Computational Methods in Applied Mechanics and Engineering 84, 327–348.

I. Christie, D. F. Griffiths, A. R. M., Zeinkiewicz, O. C., 1976. Finite Element Methods for Second Order Differential Equations with Significant First Derivatives. International Journal of Numerical Methods in Engineering 10, 1389–1396.

Jamet, P., 1978. Galerkin-Type Approximations which are Discontinuous in Time for Parabolic Equations in a Variable Domain. SIAM Journal of Numerical Analysis 15, 912–928.

Kreyszig, E., 1993. Advanced Engineering Mathematics. John Wiley & Sons, Inc.

Larsen, E. W., Morel, J. E., 1989. Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes II. J. Comp. Phys. 83, 212–236.

Larsen, E. W., Morel, J. E., W. F. Miller, J., 1987. Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes. J. Comp. Phys. 69, 283–324.

Lathowers, D., 2011. Spatially Adaptive Eigenvalue Estimation for the S_N Equations on Unstructured Triangular Meshes. Annals of Nuclear Energy 38, 1867–1876.

Lathrop, K. D., 1971. Remedies for Ray Effects. Nulcear Science And Engineering 55, 255–268.

LeSaint, P., Raviart, P. A., 1974. Mathematical Apsects of Finite Element in Partial Differential Equations, 89–123.

Lewis, E. E., Miller, W. F., 1993. Computational Methods of Neutron Transport. American Nuclear Society, Inc.

Lomtev, I., Karniadakis, G. E., 1999. A Discontinuous Galerkin Method for the Navier-Stokes Equations. International Journal of Numerical Methods in Fluids 29, 587–603.

Martin, W. R., Duderstadt, J. J., 1977. Finite Element Solutions of the Neutron Transport Equation with Applications to Strong Heterogeneities. Nuclear Science And Engineering 62, 371–390.

McClarren, R., 2011. On PN Closure, seminar given at Imperial College.

McClarren, R., Holloway, J. P., 2005. Establishing an Asymptotic Diffusion Limit for Riemann Solvers on the Time-Dependent P_N Equations. In: Proceedings of the M & C 2005 Meeting.

McClarren, R., Holloway, J. P., 2007. An Implicit Riemann Solver for the Time-Dependent P_N Equations. Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications 155(2), 290–299.

McClarren, R., Holloway, J. P., Brunner, T., Melhorn, T., 2005. An Implicit Riemann Solver for the Time-Dependent P_n Equations. Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications.

Merton, S., 2009. A non-linear optimal discontinuous petrov-galerkin method for stabilising the solution of the transport equation. In: International Conference on Mathematics, Computational Methods & Reactor Physics (M & C 2009), Saratoga Springs, New York, USA, May 3-7, 2009, on CD-ROM. American Nuclear Society (ANS), LaGrange Park, IL (2009).

Merton, S., 2010. Adjoint based error measures for functional defect correction in deterministic neutron transport applications. In: PHYSOR 2010 - Advances in Reactor Physics to Power the Nuclear Renaissance, Pittsburgh, Pennsylvania, USA, May 9-14, 2010, on CD-ROM. American Nuclear Society (ANS), LaGrange Park, IL (2009).

Merton, S., 2011. A non-linear optimal discontinuous petrov-galerkin method for stabilising the solution of the transport equation. In: International Conference on Mathematics, Computational Methods & Applied to Nuclear Science and Engineering (M & C 2011), Rio de Janeiro, RJ, Brazil, May 8-12, 2011, on CD-ROM ISBN 978-85-63688-00-2. Latin American Nuclear Society (LAS)/American Nuclear Society (ANS) (2011).

Merton, S. R., Pain, C. C., Smedley-Stevenson, R. P., Buchan, A. G., Eaton, M. D., 2008. Optimal discontinuous finite element methods for the boltzmann transport equation with arbitrary discretisation in angle. Annals of Nuclear Energy 35, 1741–1759.

Miller, W. F., Lewis, E. E., Rossow, E. C., 1973. The Application of Phase-Space Finite Elements to the Two-Dimensional Neutron Transport Equation in X-Y Geometry. Nuclear Science And Engineering 52, 12–22.

Morel, J. E., 1984. Analysis and Elimination of the Discrete-Ordinates Flux Dip. Trans. Theory Stat. Phys. 13, 615–633.

Morel, J. E., 2003. Analysis of Ray-Effect Mitigation Techniques. Nulcear Science And Engineering 144, 1–22.

Morel, J. E., Jr., J. E. D., Wareing, T. A., 1993. Diffusion-Accelerated Solution of the Two-Dimensional S_n Equations with Bilinear-Discontinuous Differencing. Nulcear Science And Engineering 115, 304–319.

Morel, J. E., Wareing, T. A., Smith, K., 1996. A Linear-Discontinuous Spatial Differencing Scheme for S_n Radiative Transfer Calculations. J. Comp. Phys. 128, 445–462.

Muller, J. D., Giles, M. B., 2001. Solution Adaptive Mesh Refinement using Adjoint Error Analysis. In: 15th Computational Fluid Dynamics Conference.

Ohnishi, T., 1971. Application of Finite Element Solution Technique to Neutron Diffusion and Transport Equations. In: Conference On New Developments In Reactor Mathematics And Applications.

Pain, C. C., Eaton, M. D., Smedley-Stevenson, R. P., 2006a. Streamline Upwind Petrov-Galerkin Methods for the Steady-State Boltzmann Transport Equation. Comp. Meth. Appl. Mech. 195, 4448–4472.

Pain, C. C., Eaton, M. D., Smedley-Stevenson, R. P., Goddard, A. J. H., Piggott, M. D., de Oliveira, C. R. E., 2006b. Space-Time Streamline Upwind Petrov-Galerkin Methods for the Boltzmann Transport Equation. Comp. Meth. Appl. Mech. Eng. 195, 4334–4357.

Pain, C. C., Goddard, A. J. H., 2000. Dual Basis and Characteristic Discontinuous Finite Element Discrtetizations for the Boltzmann Transport Equation. Transport Theory and Statistical Physics 29(6), 681–697.

Paraschivoiu, M., Patera, A., 1998. A Hierarchical Duality Approach to Bounds for the Outputs of Partial Differential Equations. Computational Methods in Applied Mechanics and Engineering 158, 389–407.

Pierce, N. A., Giles, M. B., 2000. Adjoint Recovery of Superconvergent Functionals from PDE Approximations. SIAM Review 42, 247–264.

Pierce, N. A., Giles, M. B., 2004. Adjoint and Defect Error Bounding and Correction for Functional Estimates. Journal of Computational Physics 200, 769–794.

Power, P. W., 2005. Error Measures for Finite Element Ocean Modelling. Ph.D. thesis, Imperial College London, Dept. Earth Sci. and Eng.

Reed, W. H., 1971. New Difference Schemes for the Neutron Transport Equation. Nuclear Science And Engineering 46, 309–314.

Reed, W. H., Hill, T. R., 1973. Triangular Mesh Methods for the Neutron Transport Equation. Tech. rep., Los Alamos National Laboratory.

Riesz, F., 1907. Sur une Espce de Gomtrie Analytique des Systmes de Fonctions Sommables. C. R. Acad. Sci. Paris 144, 1409–1411.

Riesz, F., 1909. Sur les Oprations Fonctionnelles Linaires. C. R. Acad. Sci. Paris 149, 974–977.

Sanchez, R., 2011. On SN-PN Equivalence. International Conference on Transport Theory 22, September 12-16 2011, Portland Oregon USA.

Seed, T. J., W. F. Miller, J., F. W. Brinkley, J., 1977. TRIDENT: A Two-Dimensional, Multigroup, Triangular Mesh Discrete Ordinates, Explicit Neutron Transport Code. Tech. rep., Los Alamos Scientific Laboratory LA-6735-M.

Sherwin, S. J., Kirby, R. M., Peiro, J., Taylor, R. L., Zienkiewicz, O. C., 2006. On 2d Elliptic Discontinuous Galerkin Methods. Int. J. Numer. Meth. Eng. 65, 752–784.

Stacey, W. M., 2007. Nuclear Reactor Physics. Wiley-VCH.

Toro, E. F., 1997. Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer, New York.

Venditti, D. A., Darfomal, D. L., 2000. Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow. Journal of Computational Physics 164, 204–227.

Venditti, D. A., Darfomal, D. L., 2002. Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows. Journal of Computational Physics 176, 40–69.

Venditti, D. A., Darfomal, D. L., 2003. Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows. Journal of Computational Physics 187, 22–46.

Warburton, T. C., Karniadakis, G. E., 1999. A Discontinuous Galerkin Method for the Viscous MHD Equations. Journal of Computational Physics 152, 1–34.

Wareing, T. A., McGhee, J. M., Morel, J. M., Pautz, S. D., 2001. Discontinuous Finite Element S_N Methods on Three-Dimensional Unstructured Grids. Nuclear Science and Engineering 138, 256–268.

Warsa, J. S., Prinja, A. K., 1999. Bilinear Discontinuous Numerical Solution of the Time Dependent Transport Equation in Slab Geometry. Annals of Nuclear Energy 26, 195–215.

Wellford, L. C., Oden, J. T., 1975. A Theory of Discontinuous Finite Element Approximations for the Analysis of Shock Waves in Nonlinear Elastic Materials. Journal of Computational Physics 19, 179–210.

Yildiz, C., 2000. P_N Solutions of the Time-Dependent Neutron Transport Equation with Anisotropic Scattering in a Homogeneous Sphere. J. Phys. D: Appl. Phys. 33, 704–710.

Zienkiewicz, O. C., 2005. The Finite Element Method its Basis & Fundamentals. Elsevier.